MiiNePort E1

User's Manual

First Edition, May 2009

www.moxa.com/product

© 2009 Moxa Inc. All rights reserved.
Reproduction without permission is prohibited.

MiiNePort E1 User's Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Copyright © 2009 Moxa Inc.
All rights reserved.
Reproduction without permission is prohibited.

Trademarks

MOXA is a registered trademark of Moxa Inc.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas:
Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa China (Shanghai office):
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-10-6872-3958

Moxa Europe:
Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa Asia-Pacific:
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

Chapter 1 Introduction 1-1
Overview 1-2
Package Checklist 1-2
Product Features 1-3
Product Specifications 1-5
Panel Layout and Pin Assignments 1-9
Panel Layout 1-9
Pin Assignments 1-9
Block Diagram 1-11
LED Indicators 1-11

Chapter 2 Getting Started 2-1
Wiring Precautions 2-2
Installing the MiiNePort E1 onto the MiiNePort E1-ST 2-3
Selecting the Serial Interface 2-3
Circuit Pad for External Connection 2-4
Connecting the Power 2-4
Connecting to the Network 2-5
Connecting to a Serial Device 2-5
Digital I/O Channel Settings 2-5
Schematic Design Guide 2-6

Chapter 3 Choosing the Proper Operation Mode 3-1
Overview 3-2
Real COM Mode 3-2
TCP Modes 3-3
TCP Server Mode 3-3
TCP Client Mode 3-4
TCP Mixed Mode 3-5
RFC2217 Mode 3-6
UDP Mode 3-7
MCSC Mode 3-8

Chapter 4 Choosing the Configuration Tool 4-1
Utility Console 4-2
Web Console 4-4
Telnet Console 4-5
SCM (Serial Command Mode) 4-6

Chapter 5 Initial IP Address Configuration 5-1
Static vs. Dynamic IP Address 5-2
Factory Default IP Address 5-2
ARP 5-2
Telnet Console 5-4

Chapter 6 Utility Console and Driver Installation 6-1
NPort Search Utility 6-2
Installing NPort Search Utility 6-2
NPort Search Utility Configuration 6-5
NPort Windows Driver Manager 6-7

Installing NPort Windows Driver Manager 6-7
Using NPort Windows Driver Manager 6-10
Installing Linux Real TTY Driver Files 6-14
Mapping TTY Ports 6-15
Removing Mapped TTY Ports 6-15
Removing Linux Driver Files 6-16

Chapter 7 Web Console Configuration 7-1
Opening Your Brower 7-2
Web Console Fundamentals 7-3
Basic Settings 7-4
Network Settings 7-4
Serial Port Settings 7-7
Operation Modes 7-9
Advanced Settings 7-34
Accessible IP List 7-34
SNMP Agent 7-35
Pin and IO Settings 7-36
Serial Command Mode (SCM) 7-37
Miscellaneous 7-38
Maintenance 7-40
Console Settings 7-40
Configuration Tool 7-40
Load Factory Default 7-41
Change Password 7-42

Chapter 8 NetEZ Technologies 8-1
SCM (Serial Command Mode) 8-2
EXTrigger (External Trigger) 8-4
AutoCFG (Auto Configuration) 8-5
MCSC (Multiple Channel Serial Communication) 8-7
Command Packets 8-10
SCM (Serial Command Mode) under MCSC 8-12

Appendix A SCM (Serial Command Mode) Command Set Introduction A-1
Command/Reply Format A-1
Command Codes for Getting Configuration A-2

Appendix B Well Known Port Numbers B-1

Appendix C Auto IP Report Protocol C-1
IP Address Report Structure C-2
Example C-2

Appendix D DIO Commands D-1
Overview D-1
C Code Example D-2
Read Single DIO D-2
Command D-2
Response D-2
C Code Example D-3
Write Single DIO D-3
Command D-3
Response D-4

C Code Example: D-4
Read Multiple DIO D-5
Command D-5
Response D-5
C Code Example: D-6
Write Multiple DIO D-7
Command D-7
Response D-8
C Code Example: D-9

Appendix E SNMP Agent with MIB II & RS-232 Like Group E-1

1

Introduction

The MiiNePort E1 Series embedded device servers are compact drop-in modules that can be integrated with your serial devices to enable connectivity to an Ethernet network. All MiiNePort E1 Series modules come equipped with built-in TCP/IP protocols and other easy-to-use network enabling tools for fast integration, allowing you to provide network access to any electronic device with a serial port.

The following topics are covered in this chapter:

- **Overview**
- **Package Checklist**
- **Product Features**
- **Product Specifications**
- **Panel Layout and Pin Assignments**
  - Panel Layout
  - Pin Assignments
- **Block Diagram**
- **LED Indicators**

Overview

Moxa's MiiNePort E1 embedded device servers are designed for manufacturers who want to add sophisticated network connectivity to their serial devices with minimal integration effort. The MiiNePort E1 is empowered by the MiiNe, Moxa's second generation SoC, which supports 10/100 Mbps Ethernet and up to 230.4 Kbps serial baudrate. MiiNePort E1 is built with a versatile selection of ready-to-use operation modes, and requires only a small amount of power. By using Moxa's innovative NetEZ technology, the MiiNePort E1 can be used to convert any device with a standard serial interface to an Ethernet enabled device in no time. In addition, the MiiNePort E1 is the size of an RJ45 connector, making it easy to fit into virtually any existing serial device.

Package Checklist

- **Modules Package**

  MiiNePort E1 (Standard Temperature Model)

  Or depending on your order;

  MiiNePort E1-T (Wide Temperature Model)

- **Starter Kits Package**
  - 1 MiiNePort E1 series module (only MiiNePort E1-ST)
  - 1 MiiNePort E1 series evaluation board
  - 1 MiiNePort E1 Series Documentation and Software CD
  - 1 universal power adaptor
  - 2 power cords
  - 1 null modem serial cable
  - 1 cross-over Ethernet cable
  - 1 product warranty booklet
  - 1 Quick Installation Guide

NOTE: Please notify your sales representative if any of the above items is missing or damaged.

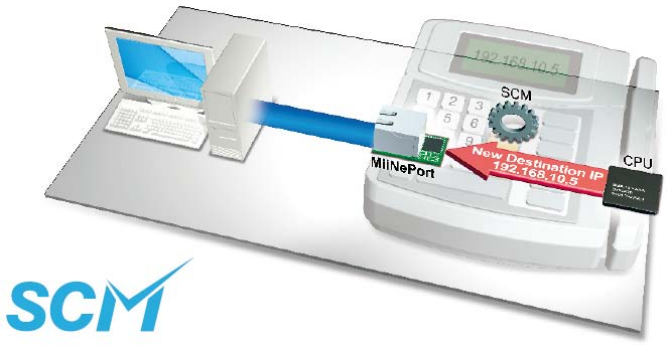
Product Features

All MiiNePort E1 Series modules have the following general features:

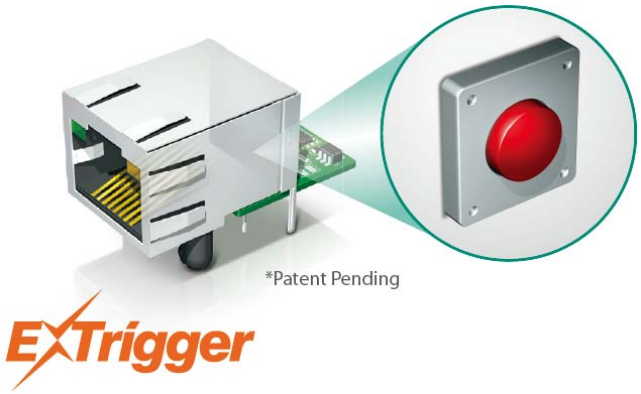
- Same size as an RJ45 connector—only 33.9 x 16.25 x 13.5 mm.
- Extremely low power consumption—only 160 mA @ 3.3 VDC input.
- Uses the MiiNe, Moxa's second generation SoC.
- NetEZ technology makes integration incredibly easy.
- Versatile choice of operation modes: Real COM, TCP, RFC2217, UDP, and MCSC.

Moxa's NetEZ technology is designed to give serial device manufacturers like you a range of powerful tools for integrating Ethernet capability into serial devices. NetEZ technology includes:

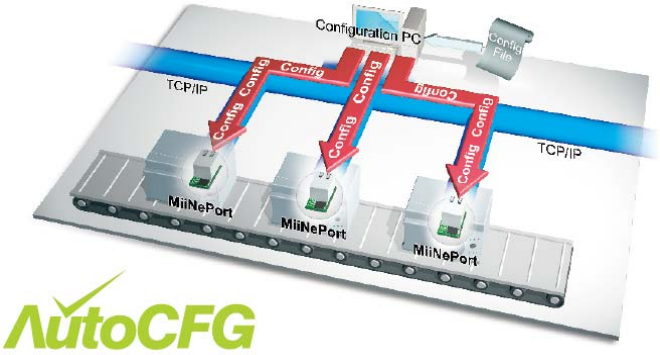
- **SCM (Serial Command Mode)** to easily configure the MiiNePort E1 through serial communication inside the device.



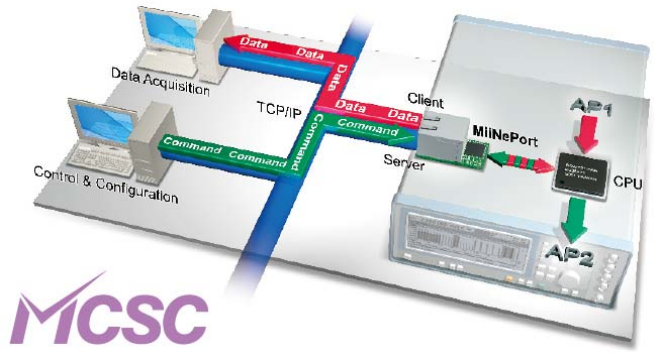
- **EXTrigger (External Trigger)** restarts the MiiNePort E1 modules or resets the modules to factory defaults with just one click for easy troubleshooting.



- **AutoCFG (Auto Configuration)** saves time and effort when setting up the MiiNePort E1 one by one during the device production process.



- **MCSC (Multiple Channel Serial Communication):** provides dual connections and dual channels so your device can act as a server and client at the same time.



Product Specifications

MiiNePort E1 Series Modules

Form Factor	
Type	Drop-in module
Dimensions	33.9 x 16.25 x 13.5 mm (13.35 x 6.4 x 5.31 in)
Weight	9 g
System Information	
CPU	32-bit ARM Core
RAM	4 MB built in
Flash	2 MB built in
Ethernet Interface	
Number of Ports	1
Speed	10/100 Mbps, auto MDI/MDIX
Connector	RJ45 (with magnetics)
Magnetic Isolation Protection	1.5 KV built-in
LEDs	10BASE-T & 100BASE-TX Link Activity, Fault/In-Use
Serial Interface	
Number of Ports	1
Transmission Format	Standard TTL
Serial Communication parameters	
Data Bits	5, 6, 7, 8
Stop Bits	1, 1.5, 2
Parity	None, Even, Odd, Space, Mark
Flow Control	RTS/CTS, XON/XOFF
Baudrate	50 bps to 230.4 Kbps* (supports non-standard baudrates) ** Baudrates up to 921.6 Kbps available by request*
Serial Signals	
TTL	TxD (Data Out), RxD (Data In), RTS, CTS, RST (reset circuit), DSR, DTR, GND
Digital I/O Pins	
GPIO	3 optional programmable I/O pins (1 DO, 1 DI, 1 DI/O)[1]
Software	
Network Protocols	ICMP, ARP, IP, TCP, UDP, DHCP, HTTP, SNMP V1/V2c, SMTP, TFTP, Auto IP, Telnet, BOOTP
Configuration Options	Web Console, Telnet Console, Windows Utility
Windows Real COM Drivers	Windows 95, 98, ME, NT, 2000, XP x86/x64, 2003 x86/x64, Vista x86/x64, 2008 x86/x64
Linux Real TTY Drivers	Linux kernel 2.4.x, 2.6.x
Operation Modes	Real COM, TCP (Server, Client, Mixed), RFC2217, UDP, and MCSC

[1] GPIO pins are Pin No. 6, 7, and 8. Please refer to Pin Assignment.

Environmental Limits

Operating Temperature:	MiiNePort E1: 0 to 55°C (32 to 131°F) MiiNePort E1-T: -40 to 85°C (-40 to 185°F)
Operating Humidity:	5 to 95% RH
Storage Temperature:	-40 to 85°C (-40 to 185°F)
Power Requirements	
Input Voltage	3.3 VDC (±5%)
Power Consumption	160 mA @ 3.3 VDC max.
Requlatory Approvals	
EMC	▪ Radiated & conducted emissions: Complies with Class B limits of EN55022:1998. ▪ Direct & Indirect ESD: Complies with EN55024:1998 ▪ Electrical Fast Transient/Burst Immunity: Complies with EN55024:1998 ▪ Power Frequency Magnetic Field Immunity: Complies with EN55024:1998
Shock	500 g's for non-operational shock
Vibration	20 g's for non-operational vibration
Automatic Reboot Trigger	Built-in WDT (watchdog timer)

MiiNePort E1-ST Evaluation Board

Power

12 to 48V DC input from power jack

Serial Interface

One DB9 connector acts as a serial data port connected to the serial device.

Serial Signals

RS-232 and RS-485 2-wire selectable for serial data port connected to the serial device

LED

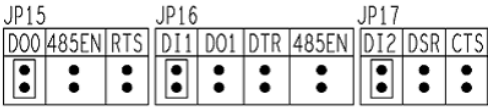
Power LED and Debug LED

Digital I/O

GPIO	3 optional programmable I/O pins (1 DO, 1 DI, 1 DI/O)
DI DIP Switch	2 DIP switch to manipulate the DI status
DO Terminal Block	2 terminal blocks to monitor the DO status

Other Peripherals

Hardware Reset Button	Hardware reset button to restart the MiiNePort E1 module
Test Board Area	Test Board area for your simulation of the real application
Configurable Pin Jumpers	JP15 for Pin No. 6, JP16 for Pin No. 7, JP17 for Pin No. 8[2]



Serial Data Port Jumpers (6 Pins)	JP19 (RS-232) and JP20 (RS-485)

ATTENTION

Users must set JP15/16/17 to correspond with the functions of Pin 6/7/8 to ensure that the Evaluation Board works properly.

ATTENTION

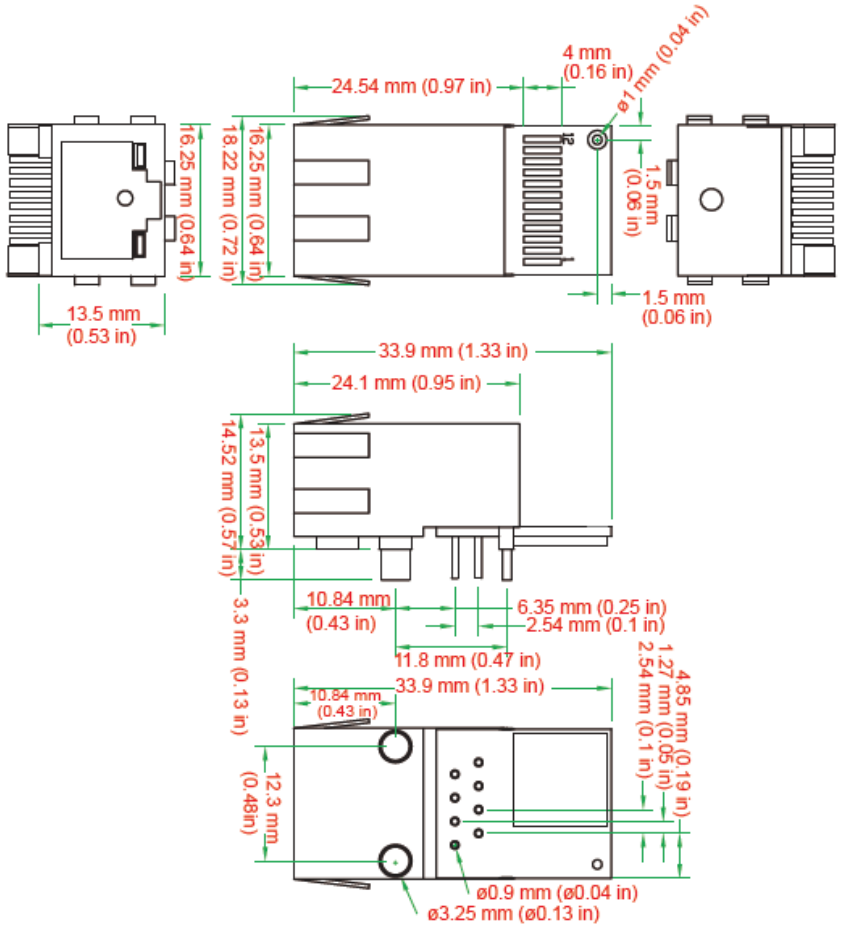
When you are in RS-485 mode, 485EN must be configured by either JP15 or JP16 and move the 6 Pin jumper from JP19 to JP20.

ATTENTION

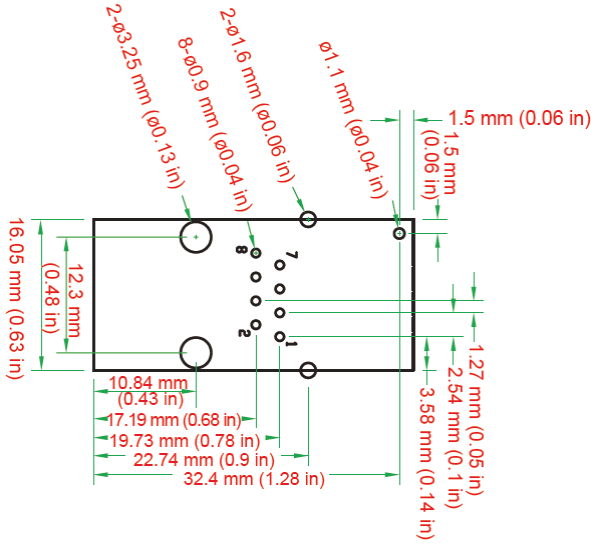
Before you manipulate the jumpers, be sure to disconnect the Power first.

[2] Configurable Pin functions (Pin No. 6, Pin No. 7, and Pin No. 8) are addressed in Pin Assignments

MiiNePort E1 Module Dimensions



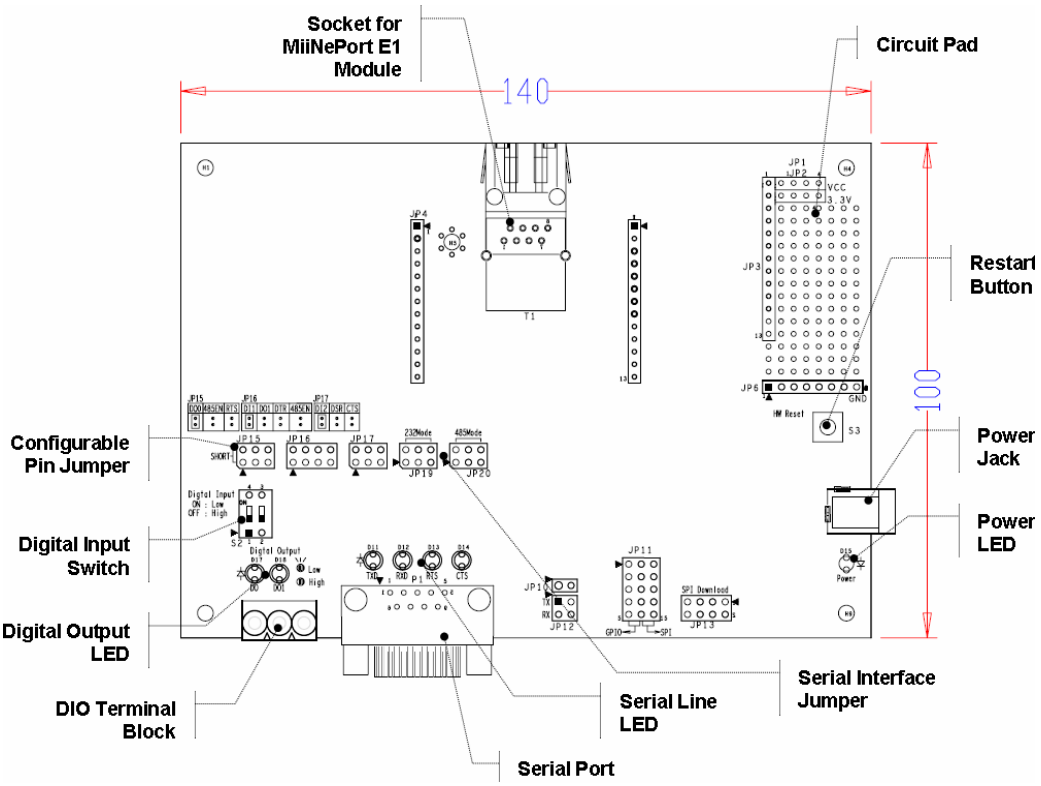
Recommended Device PCB Layout



Panel Layout and Pin Assignments

Panel Layout

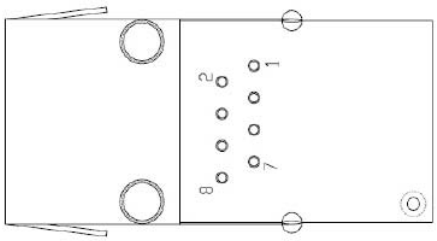
MiiNePort E1-ST Evaluation Board



Pin Assignments

MiiNePort E1 Series Modules

Serial Signal Pins



Pin	Function	Pin	Function
1	GND	5	RxD (Data In)
2	VCC	6	Ready/RTS[a]
3	Reset	7	Reset to Default[b]
4	TxD (Data Out)	8	CTS[c]

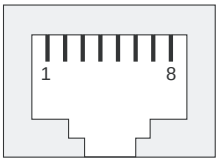
a. Pin 6 can be configured as Ready/RTS (Request to Send), Ready/DO, or RS-485 Tx Enabled (default is Ready/RTS)

b. Pin 7 can be configured as Reset to Default, DIO, DTR, or RS-485 Tx Enable (default is Reset to Default)

c. Pin 8 can be configured as CTS (Clear to Send), DI, or DSR (default is CTS)

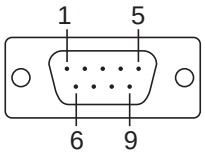
Ethernet Port Pins

Pin	Signal
1	Tx+
2	Tx-
3	Rx+
6	Rx-



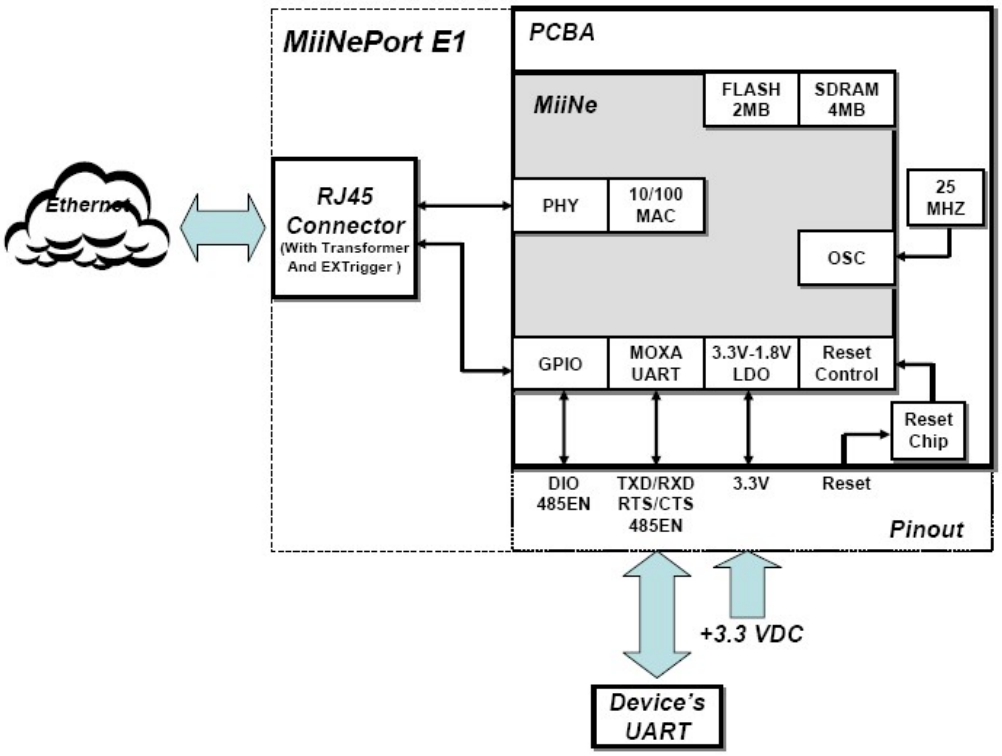
MiiNePort E1-ST Evaluation Board

DB9 Male



Pin	RS232	RS-485-2wire Signal
1	---	---
2	RxD	---
3	TxD	Data+
4	DTR	Data-
5	GND	GND
6	DSR	---
7	RTS	---
8	CTS	---

Block Diagram



LED Indicators

MiiNePort E1 Series Modules

LED	Color	Meaning
Left	Green	100BASE-TX Link Activity (constant on when link existed, blinks when data transmitting)
	Amber	10BASE-T Link Activity (constant on when link existed, blinks when data transmitting)
Right	Green	In Use (constant on when connection established)
	Amber	Fault (blinks when IP fault)

MiiNePort E1-ST Evaluation Board

Power LED

Power LED (D15) to show the status of power input.

Serial Signal LED

LED D11 to show the status of TxD

LED D12 to show the status of RxD

LED D13 to show the status of RTS

LED D14 to show the status of CTS

Digital Output LED

LED D17 to show the DO0 (Pin No. 6) status and D18 to show the DO1 (Pin No. 7) status.

2

Getting Started

This chapter includes information about how to install MiiNePort E1 Series modules for development and testing.

The following topics are covered:

- **Wiring Precautions**
- **Installing the MiiNePort E1 onto the MiiNePort E1-ST**
- **Selecting the Serial Interface**
- **Circuit Pad for External Connection**
- **Connecting the Power**
- **Connecting to the Network**
- **Connecting to a Serial Device**
- **Digital I/O Channel Settings**
- **Schematic Design Guide**

Wiring Precautions

This section describes some important safety precautions that you should pay attention to before proceeding with any installation.

ATTENTION

Be sure to disconnect the power cord before installing or wiring the evaluation board.

ATTENTION

Determine the maximum possible current in each power wire and common wire. Observe all electrical codes dictating the maximum current allowable for each wire size.

If the current goes above the maximum ratings, the wiring could overheat, causing serious damage to your equipment.

ATTENTION

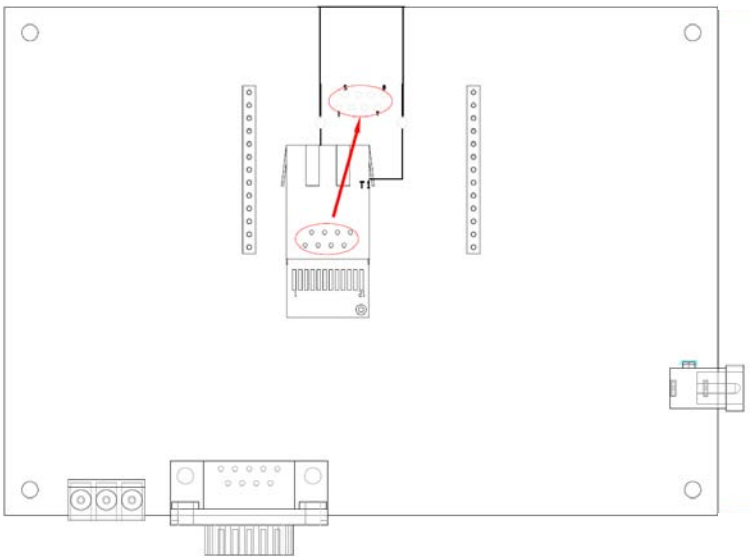
Please take care when handling the evaluation boards. When plugged in, the evaluation boards' internal components generate heat, and consequently the board may feel hot to the touch.

You should also pay attention to the following:

- Do not run signal or communication wiring and power wiring in the same wire conduit. To avoid interference, wires with different signal characteristics should be routed separately. Separate paths should be used to route wiring for power and devices. You can use the type of signal transmitted through a wire to determine which wires should be kept separate. The rule of thumb is that wires sharing similar electrical characteristics may be bundled together.
- Keep input wiring and output wiring separate.
- If power wiring and device wiring paths must cross paths, make sure the wires are perpendicular at the intersection point.
- All wiring should be clearly labeled.

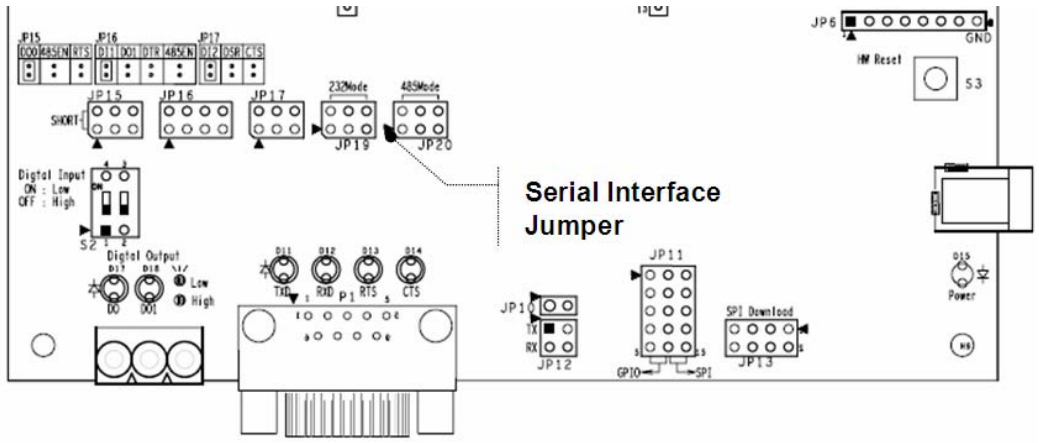
Installing the MiiNePort E1 onto the MiiNePort E1-ST

Before using the MiiNePort E1-ST evaluation board with the module, be sure to disconnect the power supply, network, and serial device. In the top center of the evaluation board, there is a MiiNePort E1 profile that indicates where you should install the module onto the evaluation board (shown in the below figure). When attaching the module to the evaluation board, make sure the 8 pins on the module is securely plugged into the 8 pin headers on the evaluation board. After the module is installed, connect the power supply, network, and serial device to the evaluation board.



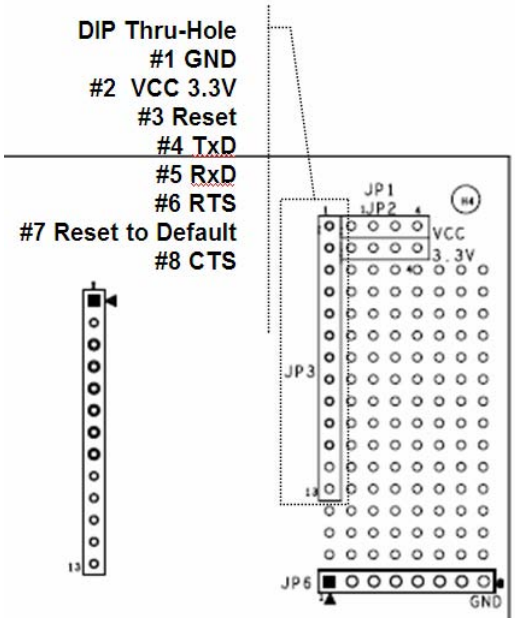
Selecting the Serial Interface

The MiiNePort E1 module comes with standard TTL serial signal input, but for your evaluation environment, the MiiNePort E1-ST has an RS-232 and RS-485 interface for you to choose from. Depending on the serial interface of the device to be connected to the evaluation board, you can choose between RS-232 and RS-485 by putting a 6-pin jumper on the evaluation board; JP19 for RS-232, J20 for RS-485 as shown in figure below.



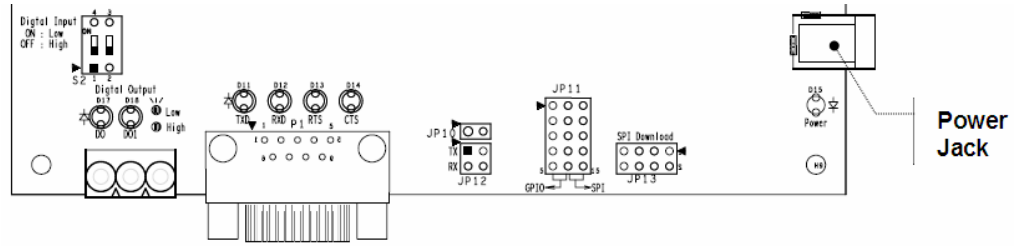
Circuit Pad for External Connection

A circuit pad is provided on the right side of each evaluation board for the development of additional application circuits.



Connecting the Power

Connect the 12~48 VDC power line with the evaluation board's power jack. If the power is properly supplied, the power LED (D15, as shown in below figure) on the evaluation board will show a solid red color until the system is ready, at which time the ready LED on the module will show a solid green color.



Connecting to the Network

To connect to the network for testing and development purposes, the module should be installed onto evaluation board. Make sure that the module is correctly installed onto the evaluation board, then plug the Ethernet cable into the RJ45 jack on the module.

If the cable is properly connected, the LED on the module (as shown in below figure) will indicate a valid connection to the Ethernet as follows:

LED	Color	Meaning
Left	Green	100BASE-TX Link Activity (constant on when link existed, blinks when data transmitting)
	Amber	10BASE-T Link Activity (constant on when link existed, blinks when data transmitting)
Right	Green	In Use (always on when connection established)
	Amber	Fault (blinks when IP fault)

When using a private IP address for the module, which is the factory default, make sure that netmask and IP settings are configured appropriately in order to access the module from a host on the network.

Connecting to a Serial Device

To connect to a serial device for testing and development purposes, the module should be installed onto the evaluation board. Make sure the serial interface is correctly chosen before connecting the evaluation board to the serial device[3]. The module's serial signals are routed to and from the RS-232 or RS-485 COM port on the evaluation board. Use a serial data cable to connect the serial device to the COM port on the evaluation board.

Digital I/O Channel Settings

Each module provides 3 digital I/O (DIO) channels[4]. All 3 DIO channels may be configured by software. A DI channel is a channel that is operating in digital input mode; a DO channel is a channel that is operating in digital output mode. You may use the evaluation board's Digital Output LEDs and Digital Input DIP switches as the digital input and output devices, or you may connect digital input/output devices to the DI/O Terminal Block.

[3] Refer to Selecting the Serial Interface when you are using jumper blocks to select the serial interface on the evaluation board.

[4] Refer to the Pin Assignments for the module's configurable DIO pin description. Refer to Configurable Pin Jumpers to select corresponding setting on the evaluation board.

ATTENTION

When using a digital input device connected to the DI/O Terminal Block, the corresponding Digital Input DIP switch must be set to "OFF" or "High". Setting the DIP switch to "ON" or "Low" will interfere with the signal from your digital input device.

For channels in digital output mode, "Low" and "High" status is controlled from within the web console. When using a Digital Output LED as your output device, "Low" status will be expressed by the LED lighting up, and "High" status will be expressed by the LED turning off.

Schematic Design Guide

For guidance and suggestions for integrating your device's hardware with the MiiNePort E1, please refer to the MiiNePort E1 Schematic Design Guide in our Document and Software CD.

3

Choosing the Proper Operation Mode

In this section, we describe the various operation modes of MiiNePort E1 modules. Depending on your intended use, you can choose the best operation mode for serial communication over the network. There is an operation mode for COM port mapping from the host computer as well as operation modes for TCP/IP protocols. After choosing the proper operating mode in this chapter, please refer to subsequent chapters for details on configuration parameters.

The following topics are covered in this chapter:

- **Overview**
- **Real COM Mode**
- **TCP Modes**
  - TCP Server Mode
  - TCP Client Mode
  - TCP Mixed Mode
- **RFC2217 Mode**
- **UDP Mode**
- **MCSC Mode**

Overview

MiiNePort E1 modules act as a bridge to connect your serial devices to the Ethernet. The built-in TCP/IP stack means less time spent on programming networking protocols. With one step you can choose the proper operation mode, then use your computer to access, manage, and configure your serial devices from anywhere in the world over the Internet.

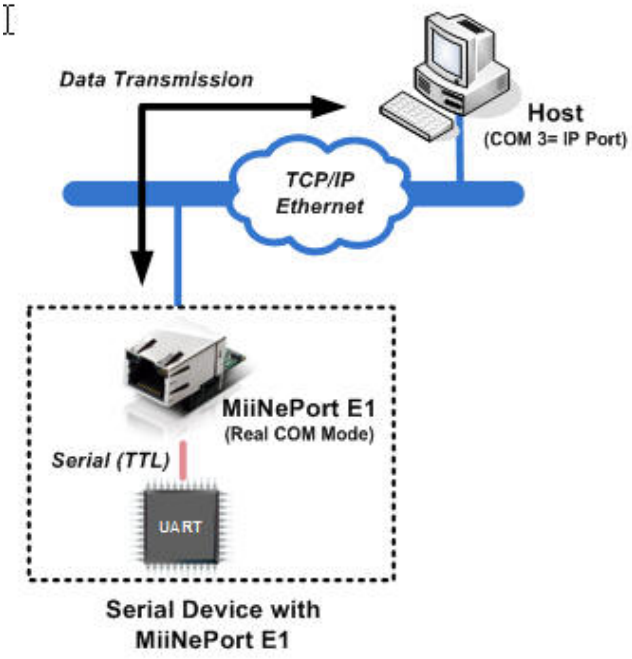
Traditional SCADA and data collection systems rely on serial ports (RS-232/422/485) to collect data from various kinds of instruments. Since MiiNePort E1 modules convert between serial and Ethernet signals, your SCADA and data collection system can be made accessible from every device connected to a standard TCP/IP network, regardless of whether the devices are used locally or at a remote site.

Real COM mode and four different socket modes, TCP Server, TCP Client, TCP Mixed, and UDP Server/Client are available on MiiNePort E1 modules. The main difference between the TCP and UDP protocol is that TCP guarantees delivery of data by requiring the recipient to send an acknowledgement to the sender.

UDP does not require this type of verification, making it possible to offer speedier delivery. UDP also allows multicasting of data to groups of IP addresses.

Real COM Mode

Real COM mode allows users to continue using software that was written for pure serial communications applications. Each module comes equipped with COM drivers for Windows systems (95 and above). The module's serial port is mapped by the driver to an IP address and port number. The driver intercepts data sent to the host's COM port, packs it into a TCP/IP packet, and then redirects it through the host's Ethernet card. At the other end of the connection, the module accepts the Ethernet frame, unpacks the TCP/IP packet, and then transparently sends the data to the attached serial device. The driver thus establishes a transparent connection between the host and serial device, allowing the host to treat the networked device as if it were directly attached.

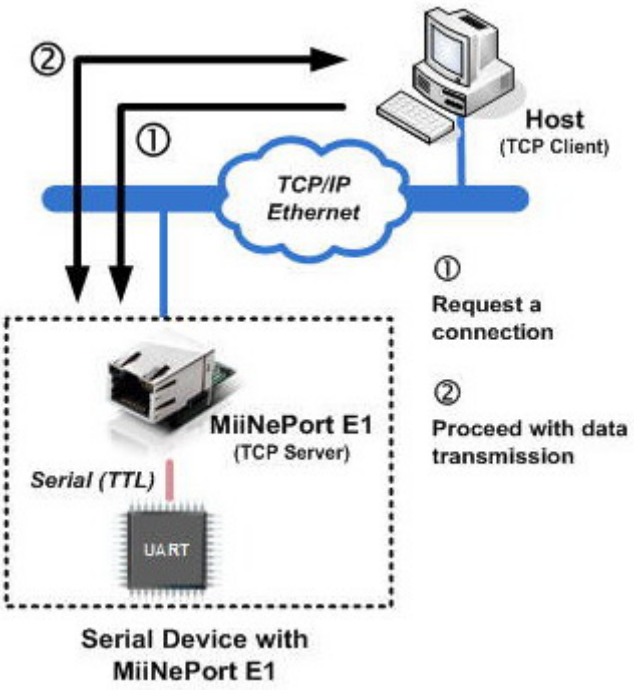


TCP Modes

TCP Server Mode

In **TCP Server mode**, the module is assigned a unique IP address and port number on the TCP/IP network. The module waits passively to be contacted by the host computer, allowing the host computer to establish a connection with and obtain data from the serial device. This operation mode also supports up to 4 simultaneous connections, so that multiple hosts can collect data from the same serial device—at the same time. As illustrated in the figure, data transmission proceeds as follows:

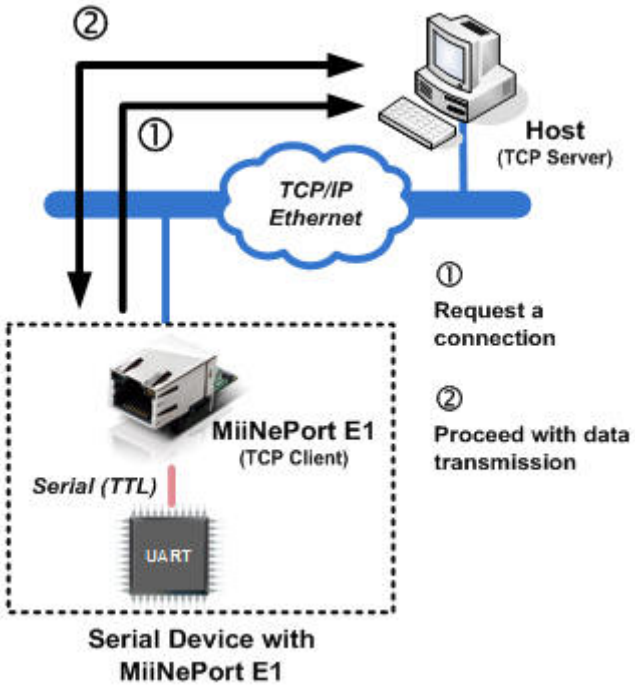
1. The host connects to the module configured for TCP Server mode.
2. Once the connection is established, data can be transmitted in both directions—from the host to the module, and from the module to the host.



TCP Client Mode

In TCP Client mode, the module can actively establish a TCP connection to a pre-defined host computer when serial data arrives. After the data has been transferred, the module can be automatically disconnected from the host computer by using the "TCP alive check time" or "Inactivity time" settings. Please refer to the following chapters for more details. As illustrated in the figure, data transmission proceeds as follows:

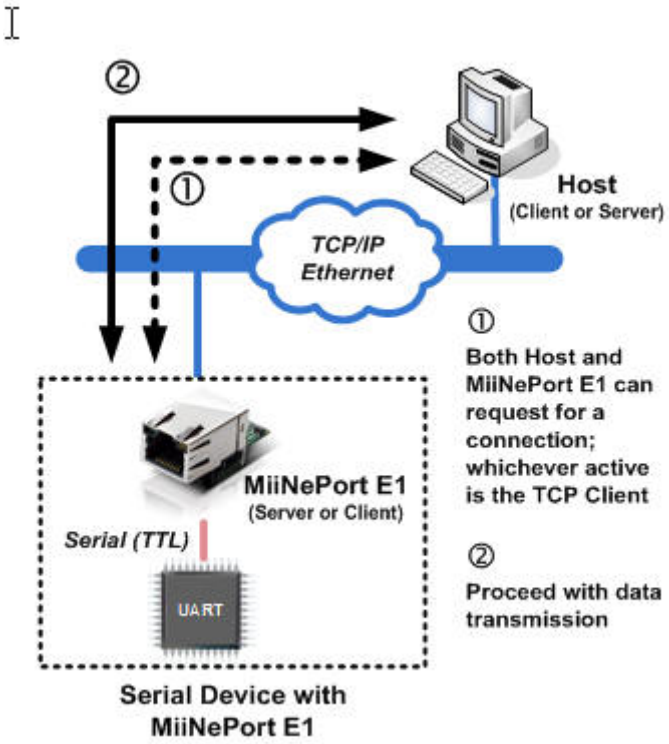
1. The module actively establishes a connection based on the conditions set in the firmware. You may let the module connect to a remote host on startup, or connect later when data from the serial device arrives.
2. Once the connection is established, data can be transmitted in both directions—from the host to the module, and from the module to the host.



TCP Mixed Mode

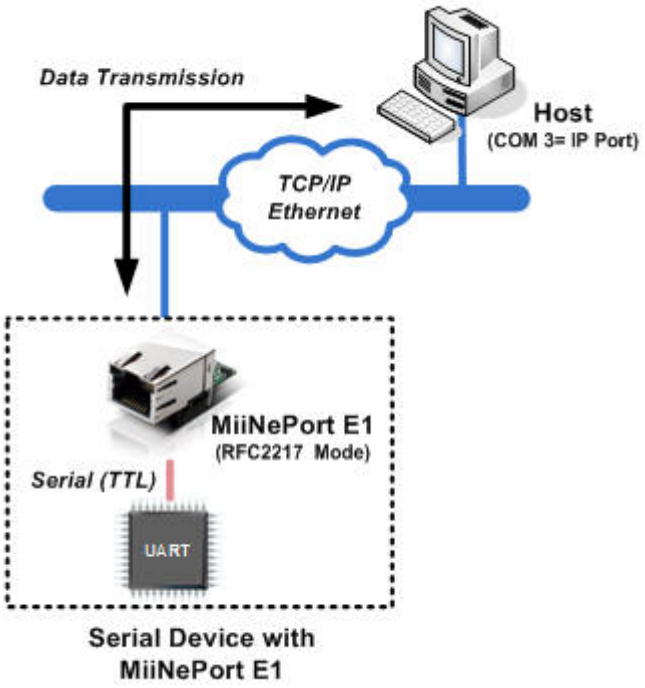
In TCP Mixed mode, the module has both TCP Server and Client settings. The MiiNePort module can act as a TCP Server or Client depending on the application. This mode is suitable for applications where the serial device actively sends data to the remote host as well as for applications where the remote host sends commands or other data to the serial device at the same time. As illustrated in the figure below, data transmission proceeds as follows:

1. Both the remote host and MiiNePort E1 can request a connection; if one acts as the TCP Client, the other acts as the TCP Server.
2. After the connection is established, data transmission will proceed.



RFC2217 Mode

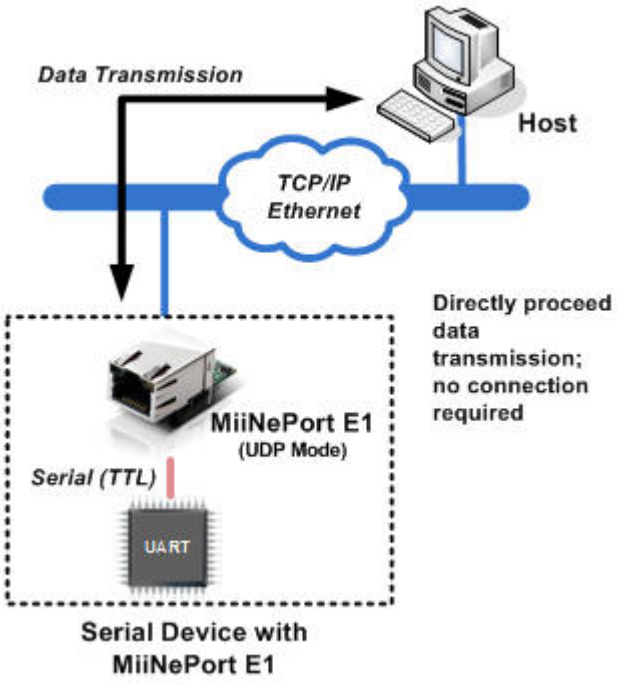
RFC2217[5] is an industrial public protocol for sharing serial devices over the TCP/IP Ethernet. Like Moxa's proprietary Real COM, RFC2217 allows users to continue using software that was written for pure serial communications applications. Each module comes equipped with COM drivers for Windows systems (95 and above). The module's serial port is mapped by the driver to an IP address and port number. The driver intercepts data sent to the host's COM port, packs it into a TCP/IP packet, and then redirects it through the host's Ethernet card.



[5] To select RFC 2217 mode, please get into TCP Server mode and set **Communication protocol** as RFC 2217. For more details, please refer to Chapter 7, Web Console Configuration.

UDP Mode

UDP mode is faster and more efficient than TCP modes. In UDP mode, you can multicast data from the serial device to multiple host computers, and the serial device can also receive data from multiple host computers, making this mode ideal for message display applications.



ATTENTION

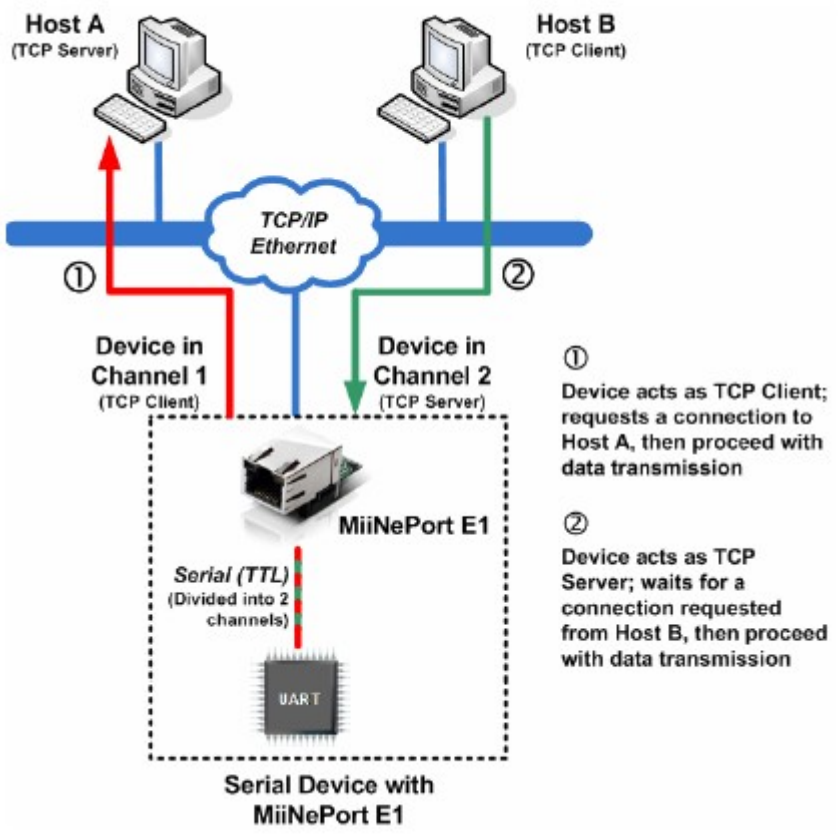
The Real COM driver comes with NPort Windows Driver Manager included in the CD-ROM in the MiiNePort E1 package.

ATTENTION

Real COM mode allows several hosts to have simultaneous access to the module. The driver controls host access to attached serial devices by checking the host's IP address against the Accessible IP list. Use the Accessible IP table to restrict access to the module when a public IP address is required for your application.

MCSC Mode

MCSC (Multiple Channel Serial Communication) was developed for multiple serial-to-Ethernet applications that use only one serial port. For example, if you wanted a solution for a scenario where your device acts as a TCP Server and TCP Client at the same time (as illustrated below), you can take advantage of MCSC.



For details of MCSC's functionality and configuration, please refer to Chapter 8, NetEZ Technologies' MCSC.

4

Choosing the Configuration Tool

MiiNePort E1 has several tools for you to configure the module. In this section we are going to briefly describe the options available and appropriate situations for those options.

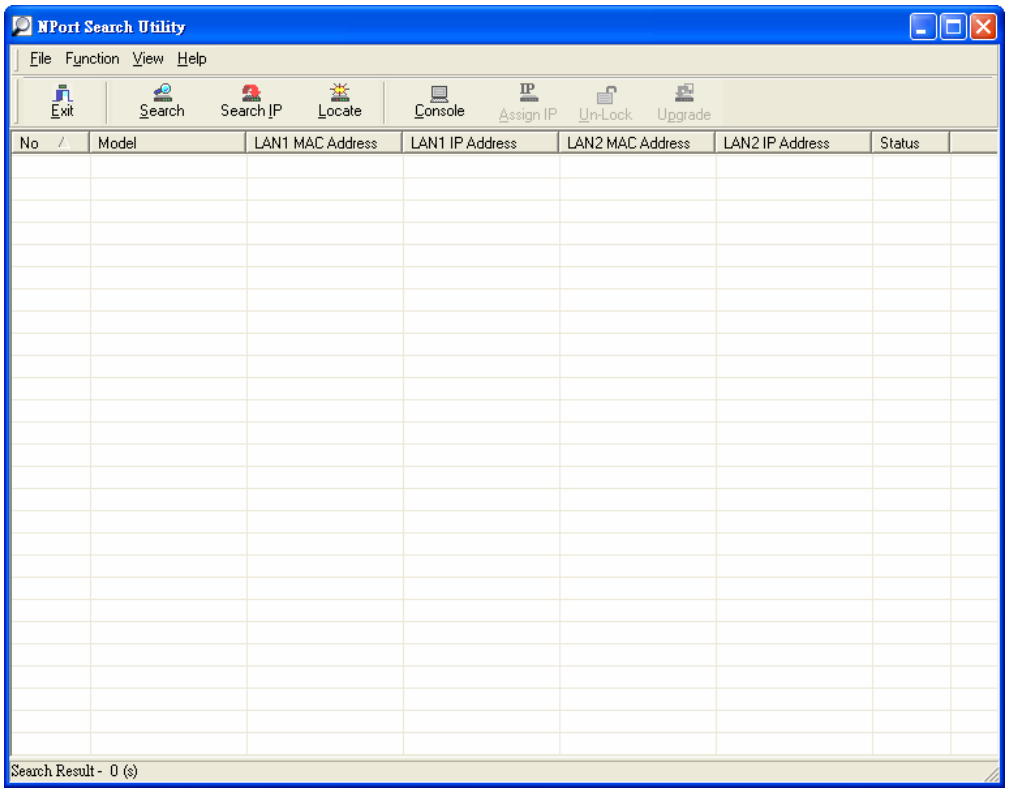
The following topics are covered:

- **Utility Console**
- **Web Console**
- **Telnet Console**
- **SCM (Serial Command Mode)**

Utility Console

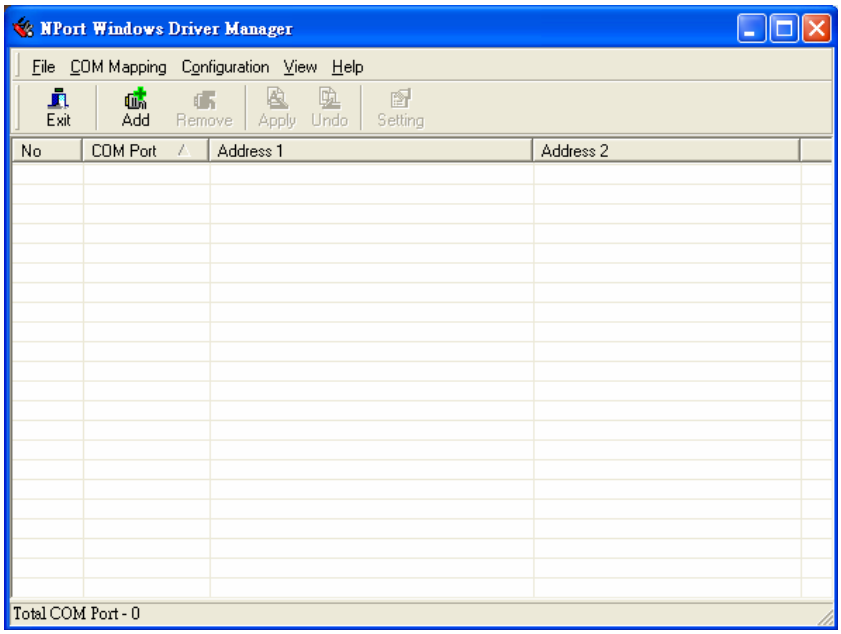
- **NPort Search Utility**

You can find NPort Search Utility in the bundled CD-ROM. NPort Search Utility is designed for Windows and is mainly used for searching the MiiNePort E1 module and assigning an IP address. You may refer to the Web Console for further configuration.



- **NPort Windows Driver Manager**

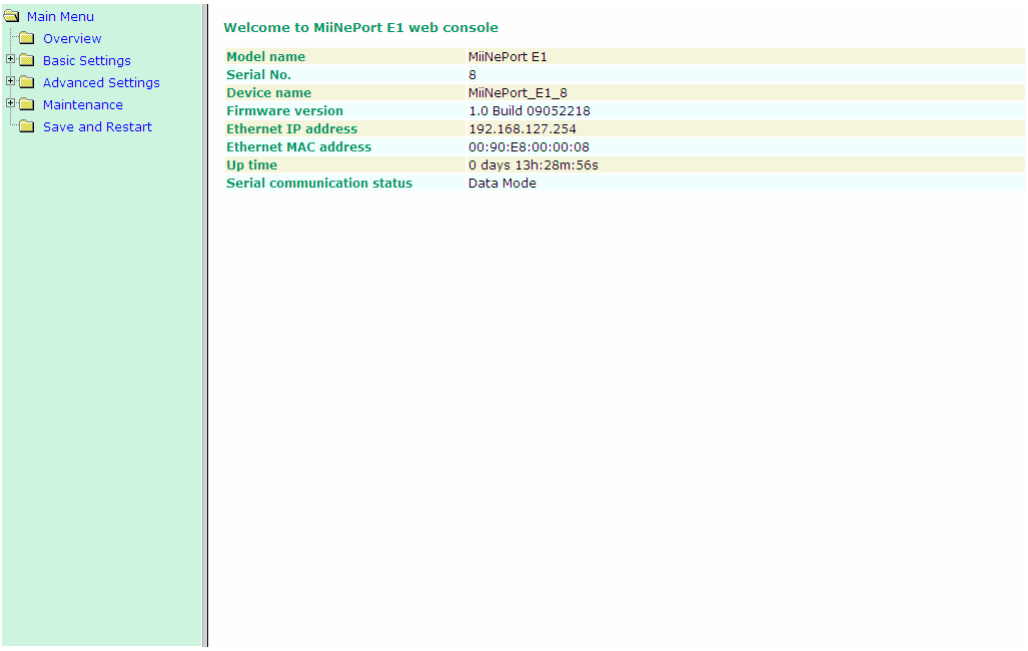
NPort Windows Driver Manager is intended for use with Real COM mode. The software manages the installation of drivers that allow you to map unused COM ports on your PC to serial ports on the MiiNePort E1.



Please refer to Chapter 6, Utility Console and Driver Installation, for details on how to use NPort Search Utility and NPort Windows Driver Manager.

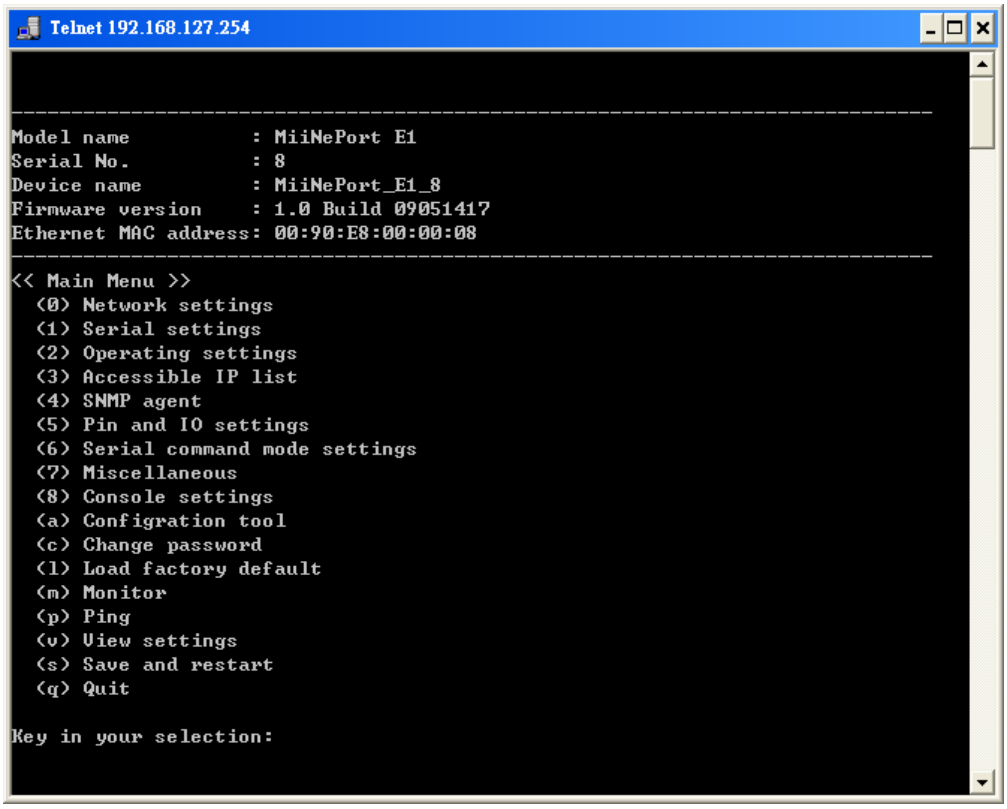
Web Console

After correctly searching the MiiNePort E1 with NPort Search Utility, you may then configure your MiiNePort E1 using a standard web browser. Please refer to Chapter 7, Web Console Configuration, for details on how to access and use the MiiNePort E1 web console.



Telnet Console

For your convenience, you may use network access to configure the MiiNePort E1 by using the Telnet program, which requires that the module have a network connection and an IP address. We briefly touch the Telnet console configuration by introducing the Initial IP Address Configuration in Chapter 5. If you want to use other configuration commands in Telnet console, please refer to Chapter 7, Web Console Configuration, for introduction of the same commands.



SCM (Serial Command Mode)

MiiNePort E1's SCM (Serial Command Mode) allows the module's parameters to be retrieved or configured through the serial port, rather than over the network. This is done through the use of specially parsed commands sent to the module through the serial port.

SCM is often used when your device has already been used in a real application and a configuration change on the MiiNePort E1 is required, such as changing the device's IP address by the operation on your device's key pad.

Please refer to Chapter 7, Web Console Configuration, for details on how to access and use the MiiNePort SCM. Please refer to Appendix A for the SCM command set instructions.

5

Initial IP Address Configuration

When setting up your MiiNePort E1 module for the first time, the first thing you should do is to configure the IP address. This chapter introduces the methods that can be used to configure the module's IP address. For more details about network settings, please refer to the Network Settings section from Chapter 7, Web Console Configuration.

This chapter includes the following sections:

- **Static vs. Dynamic IP Address**
- **Factory Default IP Address**
- **ARP**
- **Telnet Console**

Static vs. Dynamic IP Address

You should first determine whether the module will be assigned a Static IP or Dynamic IP (either DHCP or BOOTP application) in your application.

- If the module is used in a Static IP environment, you have to configure the IP address directly.
- If the module is used in a Dynamic IP environment, you have to configure the module to obtain an IP address dynamically with DHCP, DHCP/BOOTP, or BOOTP.

ATTENTION

Consult your network administrator on how to reserve a fixed IP address for the module in the MAC-IP mapping table when using a DHCP Server or BOOTP Server. For most applications, you should assign a fixed IP address to the module.

Factory Default IP Address

The MiiNePort E1 module is configured with the following default private IP address:

192.168.127.254

Please note that IP addresses of the form 192.168.xxx.xxx are referred to as private IP addresses, since it is not possible to directly access a device configured with a private IP address from a public network. For example, you would not be able to ping such a device from an outside Internet connection. Applications that require sending data over a public network, such as the Internet, require setting up the server with a valid public IP address, which can be leased from a local ISP.

ARP

You can use the ARP (Address Resolution Protocol) command to set up the module's IP address. The ARP command tells your computer to associate the module's MAC address with the intended IP address. You must then use Telnet to access the module, at which point the module's IP address will be reconfigured.

ATTENTION

In order to use ARP, both your computer and the module must be connected to the same LAN. You may also use a cross-over Ethernet cable to connect the module directly to your computer's Ethernet port. Your module must be configured with the factory default IP address before executing the ARP command.

To configure the IP address using ARP, follow these instructions:

1. Obtain a valid IP address for the module from your network administrator.
2. Obtain the module's MAC address from the label on the module.
3. Execute the 'arp -s' command from your computer's MS-DOS prompt by typing:

   arp –s <new IP address> 00-90-E8-tt-tt-tt

For example,

arp –s 192.168.200.100 00-90-E8-00-00-00

In the example above, 192.168.200.100 is the new IP address and 00-90-E8-00-00-00 is the module's MAC address, as obtained in steps 1 and 2.

4. Execute a special Telnet command by typing:

   telnet <new IP address> 6000

For example,

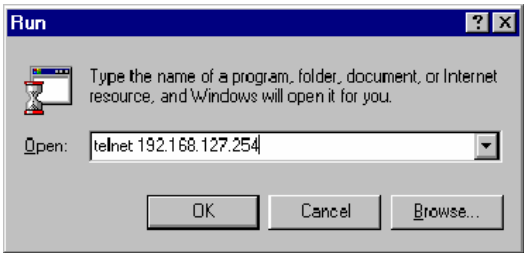
telnet 192.168.200.100 6000

After issuing this command, a **Connect failed** message will appear. After the module reboots, its IP address will be updated to the new address, and you can reconnect the module using utility, web or Telnet console to verify that the update was successful.

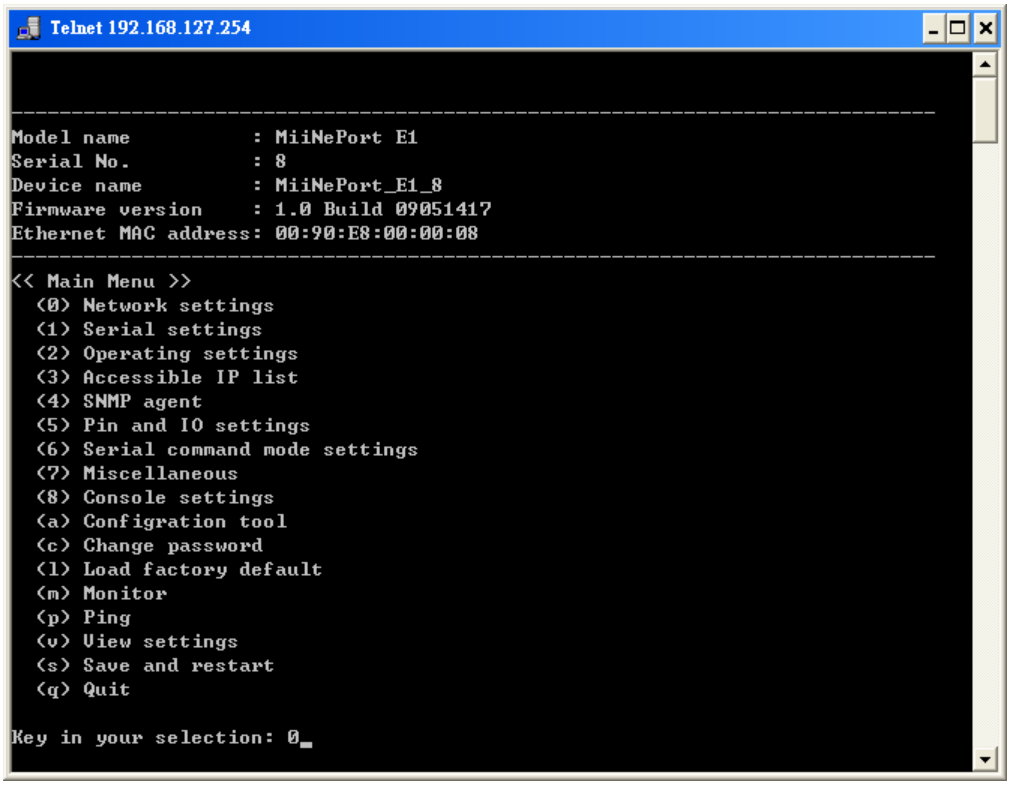
Telnet Console

Depending on how your computer and network are configured, you may find it convenient to use network access to set up your module's IP address. This can be done using the Telnet program, which requires that the module have a network connection and an IP address.

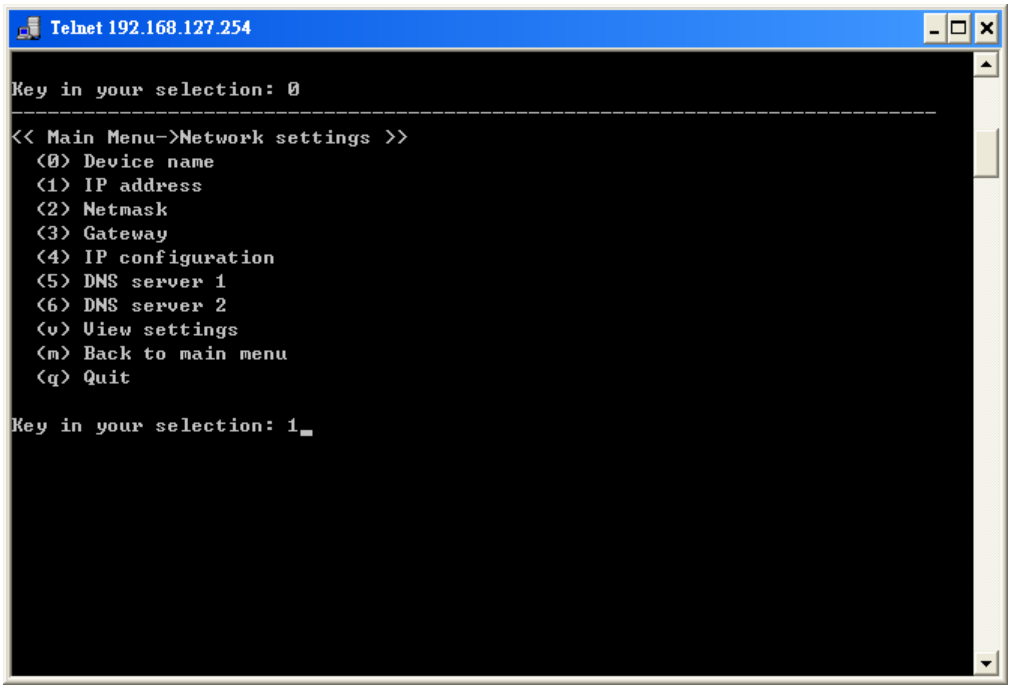
1. From the Windows desktop, click on **Start** and then select **Run**.
2. Telnet to the module's current IP address. If this is the first time configuring the module, you will telnet to the default IP address by typing **telnet 192.168.127.254** in the **Open** text box, then clicking **OK**.



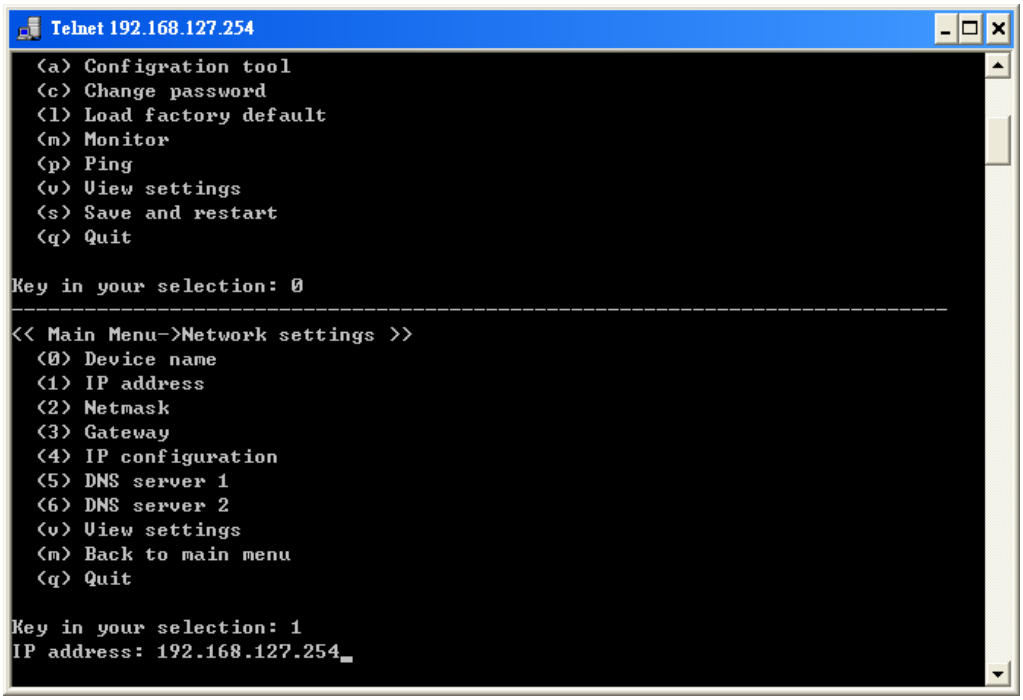
3. Select **Network settings** by pressing **0** and then **Enter**.



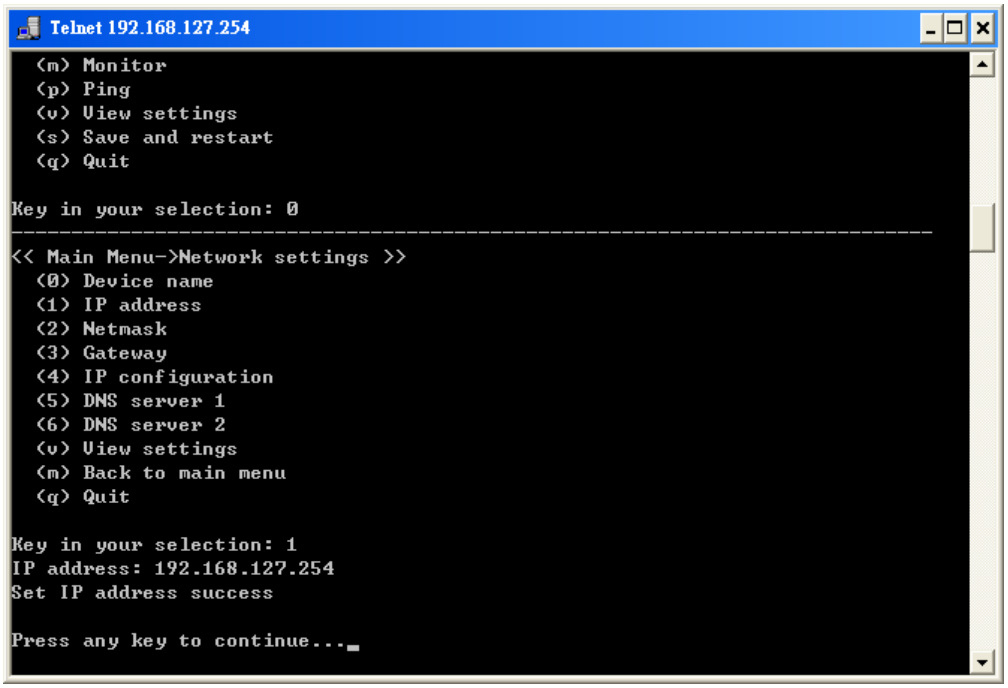
4. Select **IP address** by pressing **1** and then **Enter**.



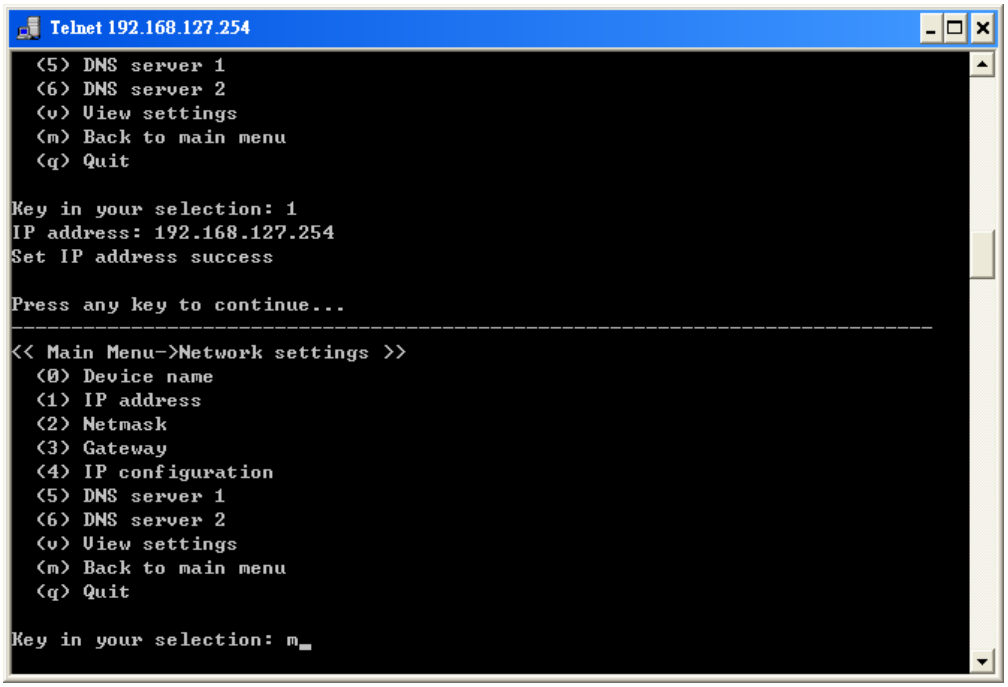
5. Use the **Backspace** key to erase the current IP address, then type in the new IP address and press **Enter**.



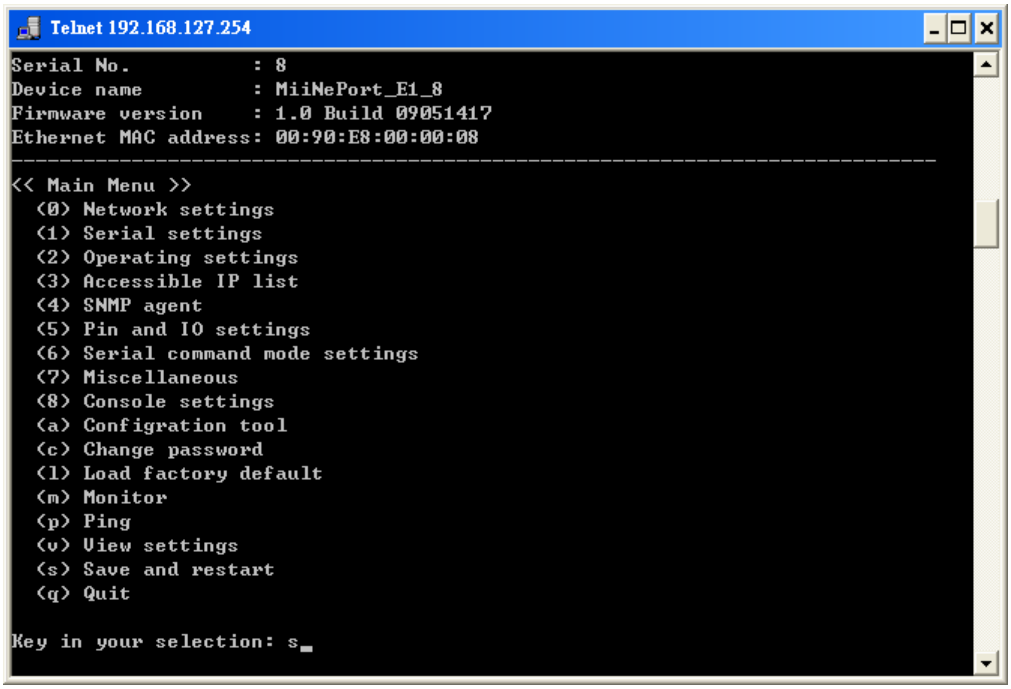
6. Press any key to continue.



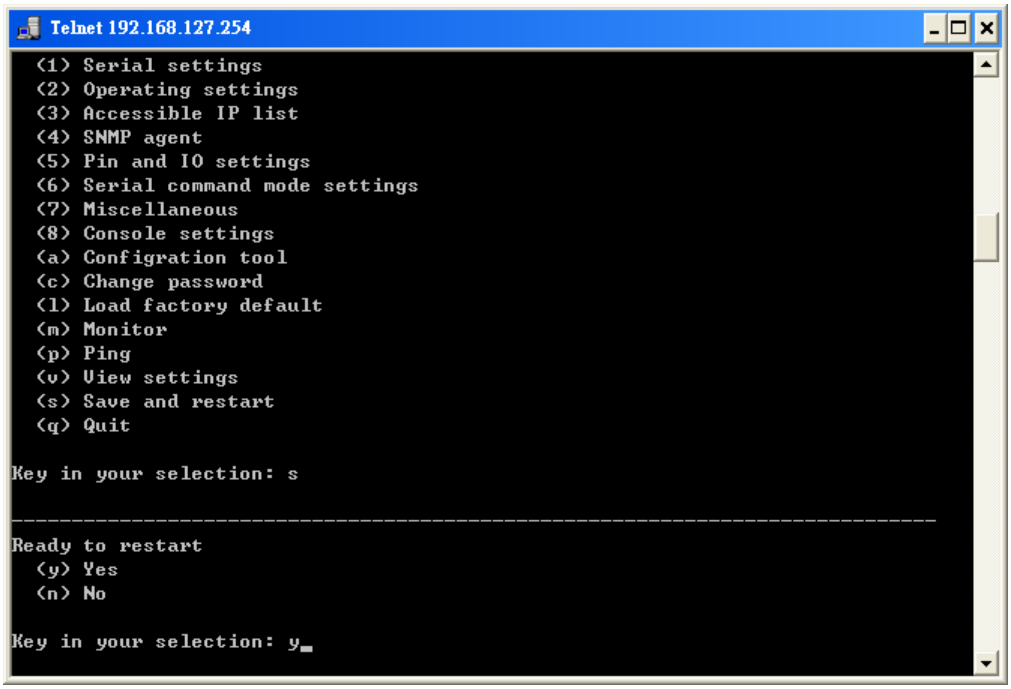
7. Press **M** and then **Enter** to return to the main menu.



8. Press **S** and then **Enter** to **Save/Restart** the system.



9. Press **Y** and then **Enter** to save the new IP address and restart the module.



6

Utility Console and Driver Installation

This chapter teaches you how to install the MiiNePort E1's utilities, use the utilities to perform simple configurations, and install the drivers.

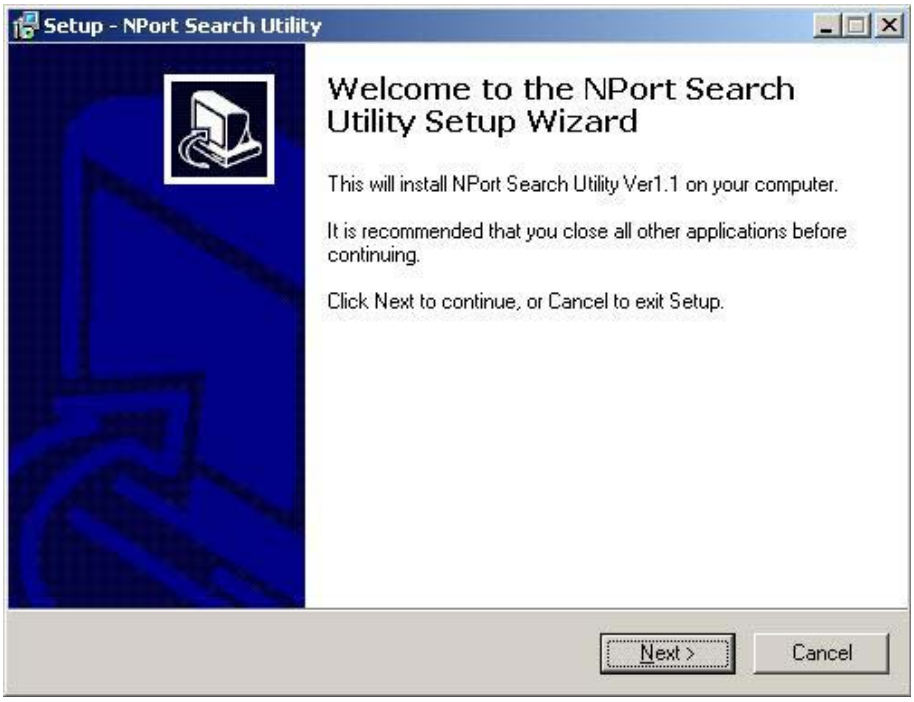
This chapter covers the following topics:

- **NPort Search Utility**
  - Installing NPort Search Utility
  - NPort Search Utility Configuration
- **NPort Windows Driver Manager**
  - Installing NPort Windows Driver Manager
  - Using NPort Windows Driver Manager
- **Installing Linux Real TTY Driver Files**
  - Mapping TTY Ports
  - Removing Mapped TTY Ports
  - Removing Linux Driver Files

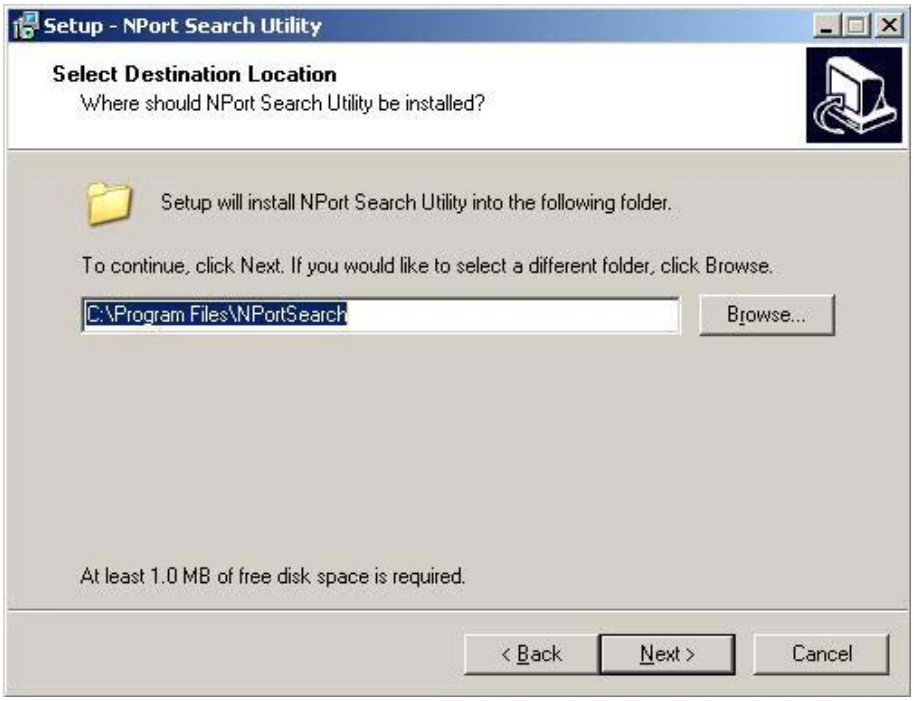
NPort Search Utility

Installing NPort Search Utility

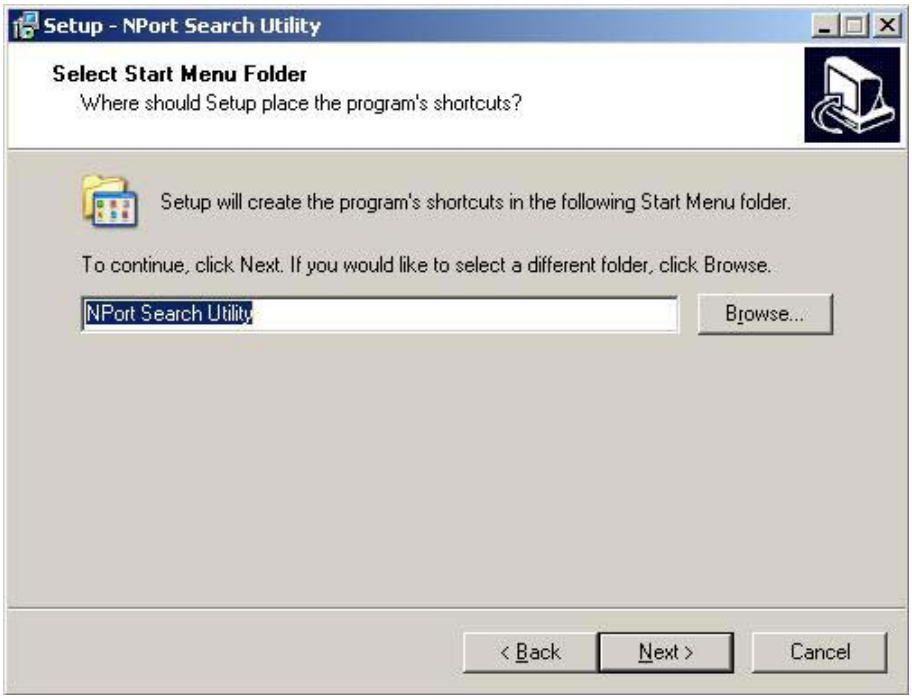
1. Click the **INSTALL UTILITY** button in the MiiNePort E1 Installation CD auto-run window to install NPort Search Utility. Once the program starts running, click **Yes** to proceed.
2. Click **Next** when the Welcome screen opens, to proceed with the installation.


Setup - NPort Search Utility
Welcome to the NPort Search Utility Setup Wizard
This will install NPort Search Utility Ver1.1 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

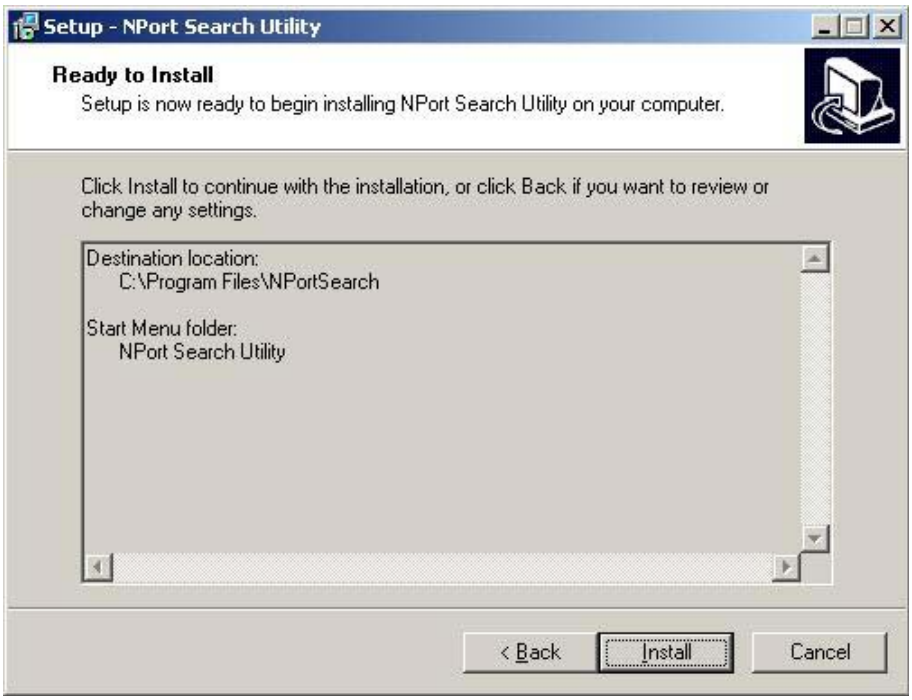
3. Click **Next** to install program files to the default directory, or click **Browse** to select an alternate location.


Setup - NPort Search Utility
Select Destination Location
Where should NPort Search Utility be installed?
Setup will install NPort Search Utility into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\NPortSearch
Browse...
At least 1.0 MB of free disk space is required.
< Back
Next >
Cancel

4. Click **Next** to install the program's shortcuts in the appropriate Start Menu folder.

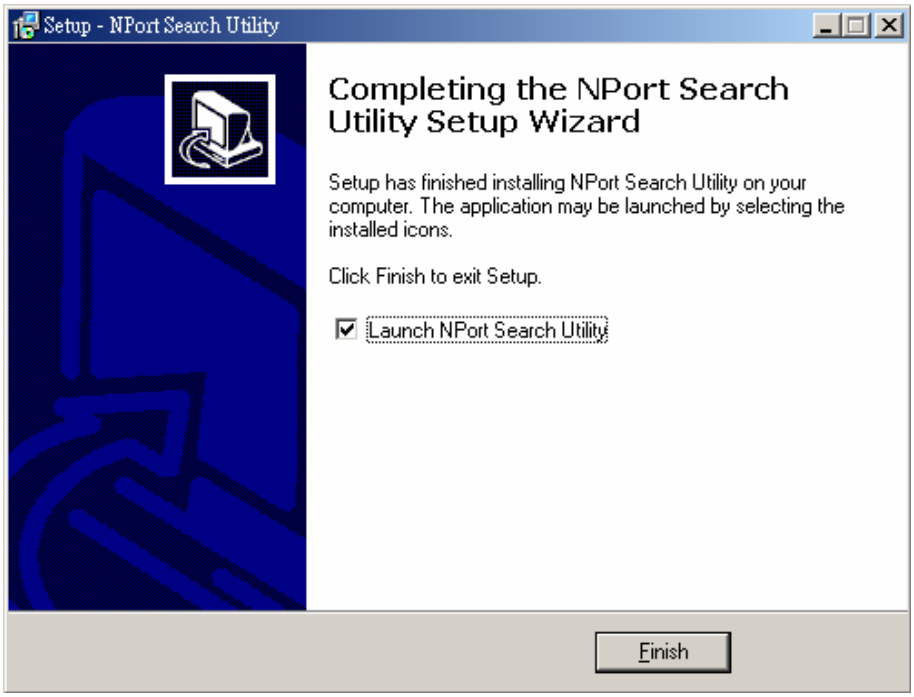

Setup - NPort Search Utility
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
NPort Search Utility
Browse...
< Back
Next >
Cancel

5. Click **Next** to proceed with the installation. The installer then displays a summary of the installation options.


Setup - NPort Search Utility
Ready to Install
Setup is now ready to begin installing NPort Search Utility on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Program Files\NPortSearch
Start Menu folder:
NPort Search Utility
< Back
Install
Cancel

6. Click **Install** to begin the installation. The setup window will report the progress of the installation. To change the installation settings, click **Back** and navigate to the previous screen.

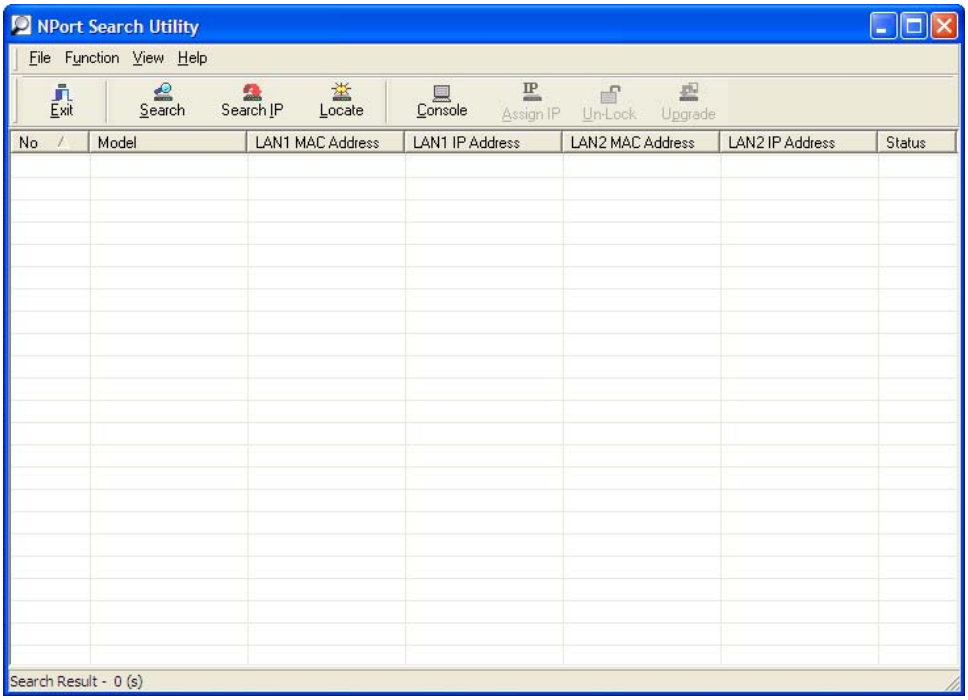
7. Click **Finish** to complete the installation of NPort Search Utility.


Setup - NPort Search Utility
Completing the NPort Search Utility Setup Wizard
Setup has finished installing NPort Search Utility on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch NPort Search Utility
Finish

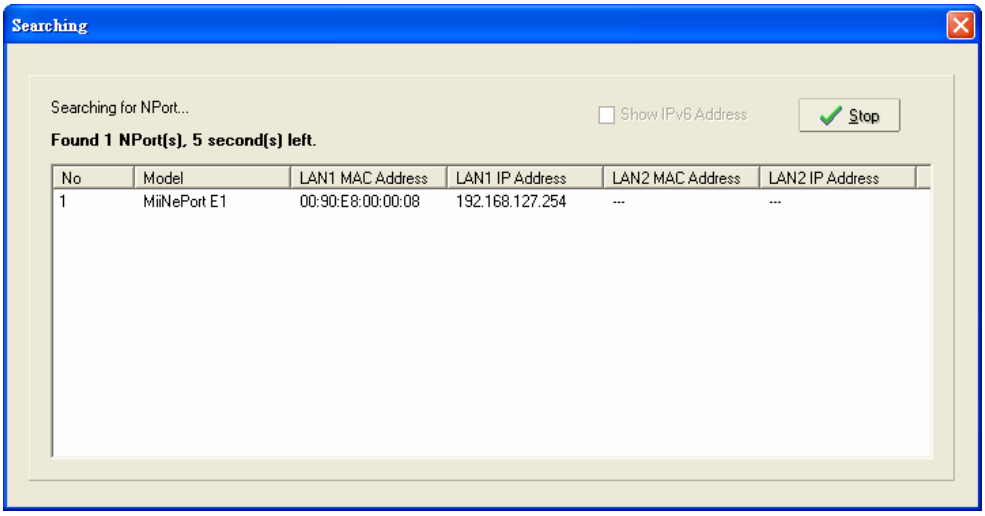
NPort Search Utility Configuration

The Broadcast Search function is used to locate all MiiNePort E1 modules that are connected to the same LAN as your computer. After locating a MiiNePort E1, you will be able to change its IP address. Since the Broadcast Search function searches by MAC address and not IP address, all MiiNePort E1 modules connected to the LAN will be located, regardless of whether or not they are part of the same subnet as the host.

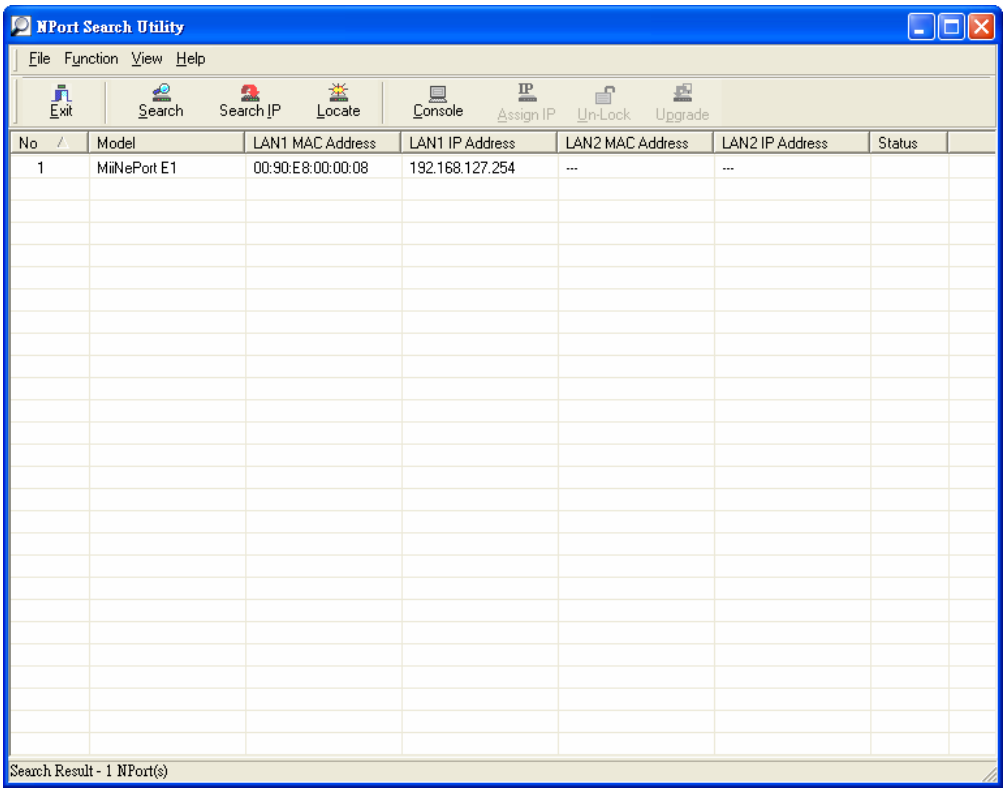
1. Open NPort Search Utility and then click the **Search** icon.



The Searching window indicates the progress of the search.



2. When the search is complete, all MiiNePort E1 modules that were located will be displayed in the NPort Search Utility window.



To modify the configuration of the highlighted MiiNePort E1, click on the **Console** icon to open the web console. This will take you to the web console, where you can make all configuration changes. Please refer to Chapter 7, Web Console Configuration, for information on how to use the web console.

ATTENTION

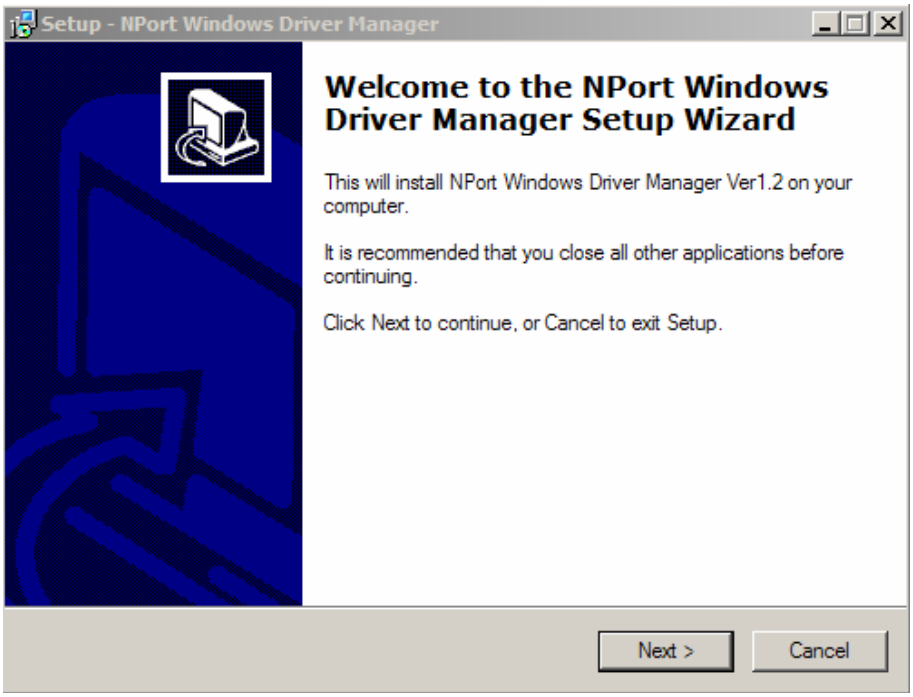
If you are looking for information related to TCP, RFC2217, or UDP modes, you can ignore the following Driver sections, including NPort Windows Driver Manager and Linux Real TTY Driver. Please jump to Chapter 7, Web Console Configuration for further settings.

NPort Windows Driver Manager

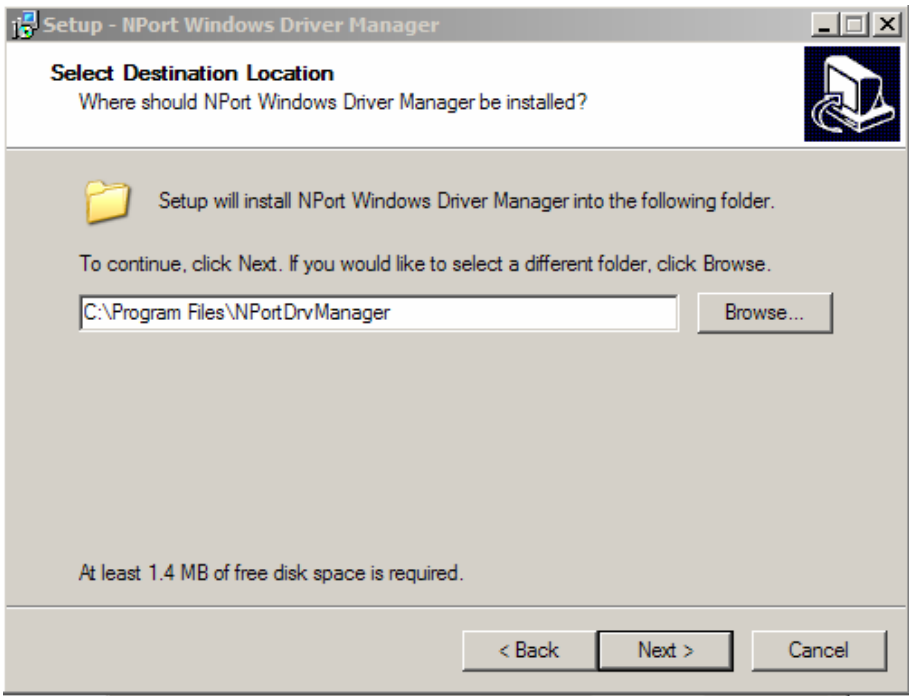
Installing NPort Windows Driver Manager

NPort Windows Driver Manager is intended for use with serial ports that are set to Real COM mode. The software manages the installation of drivers that allow you to map unused COM ports on your PC to your device presented by MiiNePort E1's serial port. These drivers are designed for use with Windows 95, 98, ME, NT, 2000, XP x86/x64, 2003 x86/x64, Vista x86/x64, 2008 x86/x64. When the drivers are installed and configured, your devices that are embedded with the MiiNePort E1 will be treated as if they were attached to your PC's own COM ports.

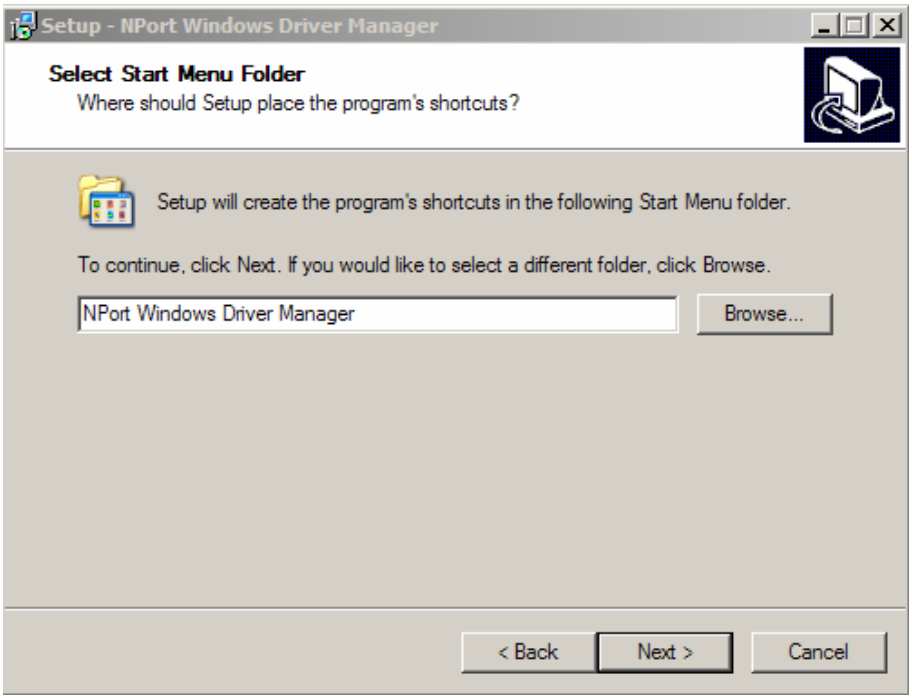
1. Click the **INSTALL COM Driver** button in the MiiNePort E1 Installation CD auto-run window to install the NPort Windows Driver. Once the installation program starts running, click **Yes** to proceed.
2. Click **Next** when the Welcome screen opens, to proceed with the installation.


Setup - NPort Windows Driver Manager
Welcome to the NPort Windows Driver Manager Setup Wizard
This will install NPort Windows Driver Manager Ver1.2 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

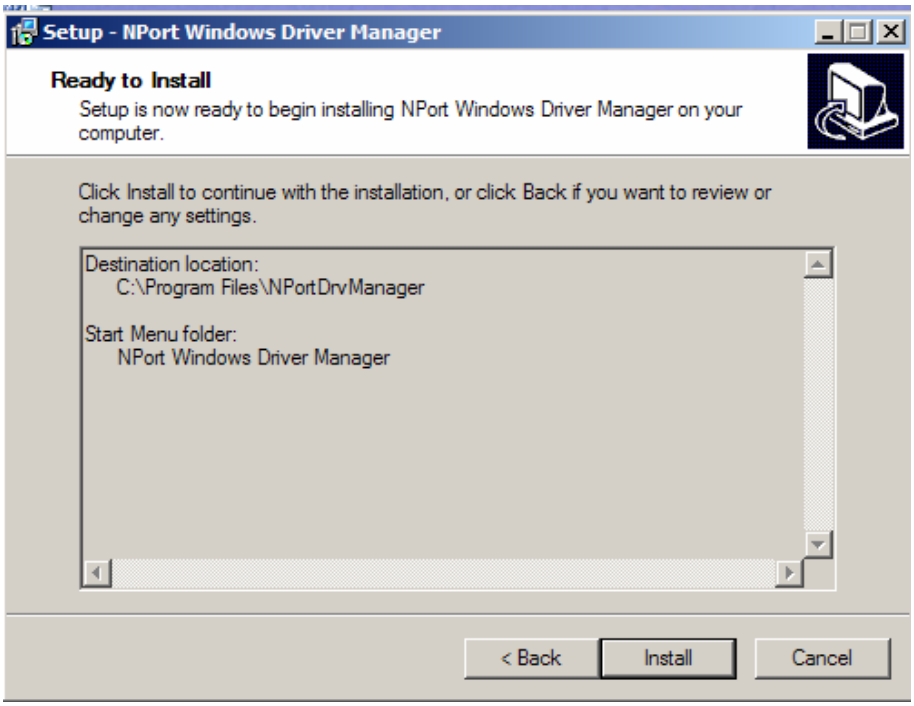
Click **Next** to install program files to the default directory, or click **Browse** to select an alternate location.


Setup - NPort Windows Driver Manager
Select Destination Location
Where should NPort Windows Driver Manager be installed?
Setup will install NPort Windows Driver Manager into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\NPortDrvManager
Browse...
At least 1.4 MB of free disk space is required.
< Back
Next >
Cancel

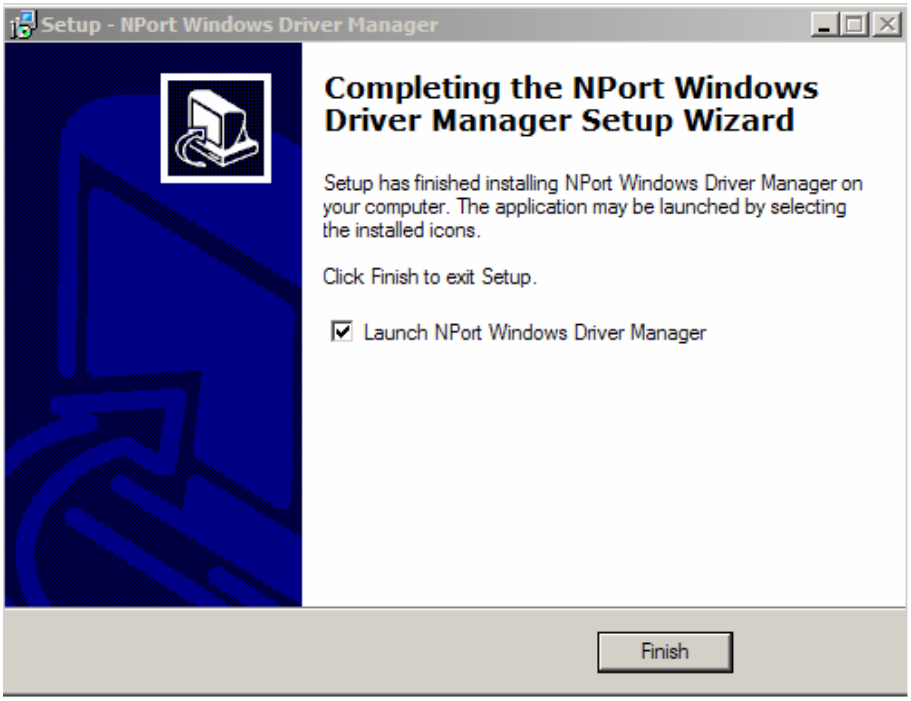
3. Click **Next** to install the program's shortcuts in the appropriate Start Menu folder.


Setup - NPort Windows Driver Manager
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
NPort Windows Driver Manager
Browse...
< Back
Next >
Cancel

4. Click **Next** to proceed with the installation. The installer then displays a summary of the installation options.


Setup - NPort Windows Driver Manager
Ready to Install
Setup is now ready to begin installing NPort Windows Driver Manager on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Program Files\NPortDrvManager
Start Menu folder:
NPort Windows Driver Manager
< Back
Install
Cancel

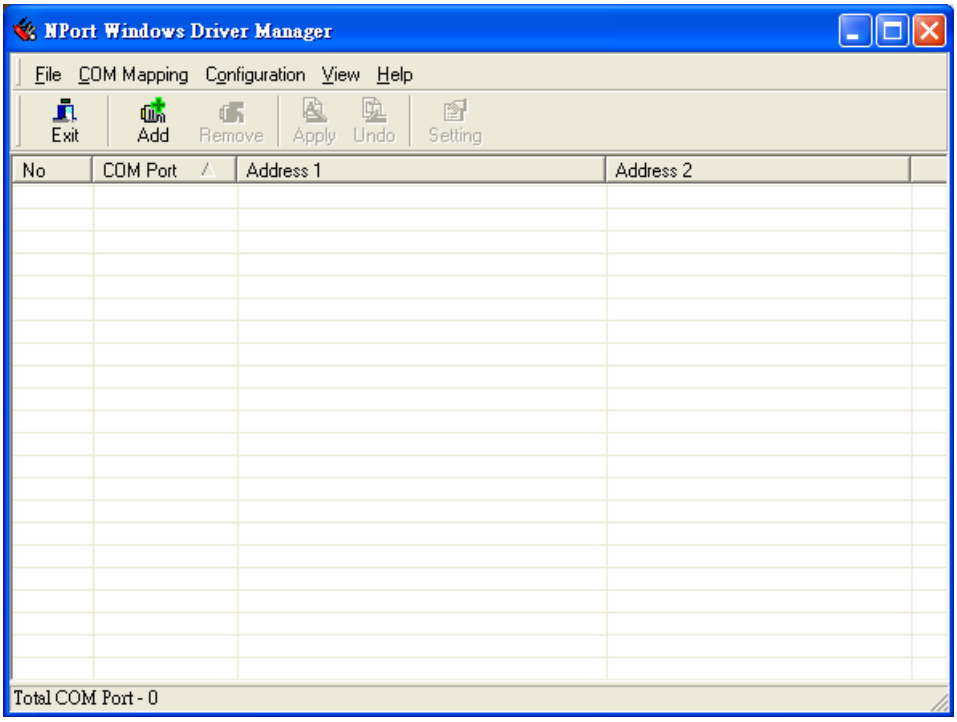
5. Click **Install** to begin the installation. The setup window will report the progress of the installation. To change the installation settings, click **Back** and navigate to the previous screen.
6. Click **Finish** to complete the installation of the NPort Windows Driver Manager.


Setup - NPort Windows Driver Manager
Completing the NPort Windows Driver Manager Setup Wizard
Setup has finished installing NPort Windows Driver Manager on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch NPort Windows Driver Manager
Finish

Using NPort Windows Driver Manager

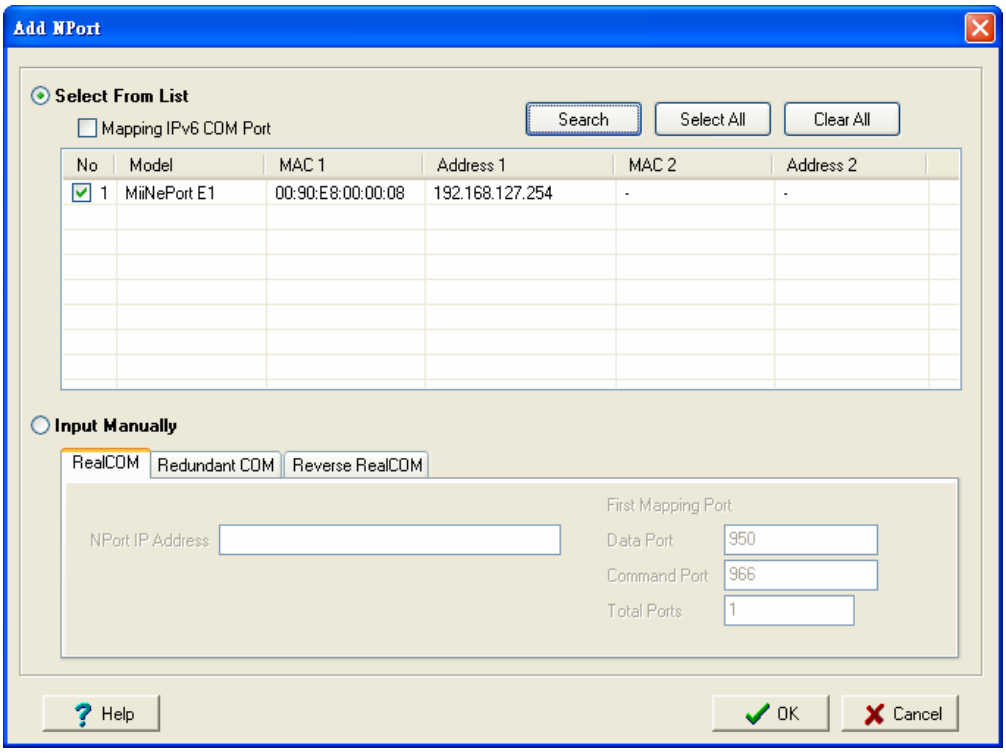
After you install NPort Windows Driver Manager, you can set up the MiiNePort E1's serial port, which is connected to your device's main board, as remote COM ports for your PC host. **Make sure that the serial port on your MiiNePort E1 is already set to Real COM mode when mapping COM ports with the NPort Windows Driver Manager** [6].

1. Go to **Start → NPort Windows Driver Manager → NPort Windows Driver Manager** to start the COM mapping utility.
2. Click the **Add** icon.

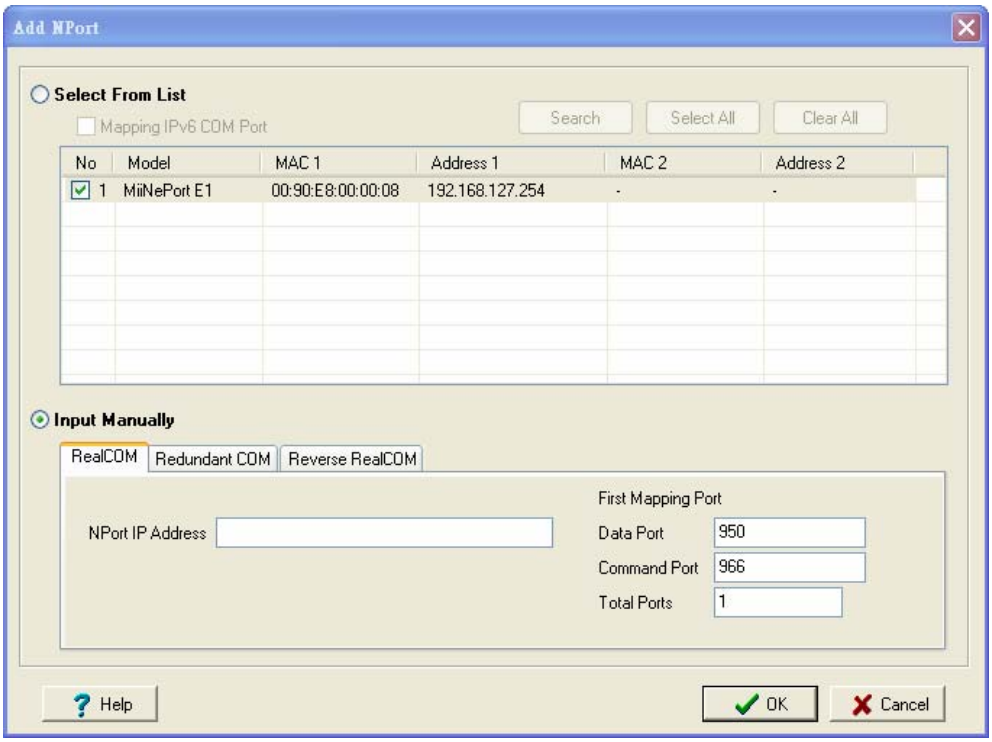


[6] Please refer to Chapter 7, Web Console Configuration, for learning how to configure your MiiNePort E1 to Real COM mode.

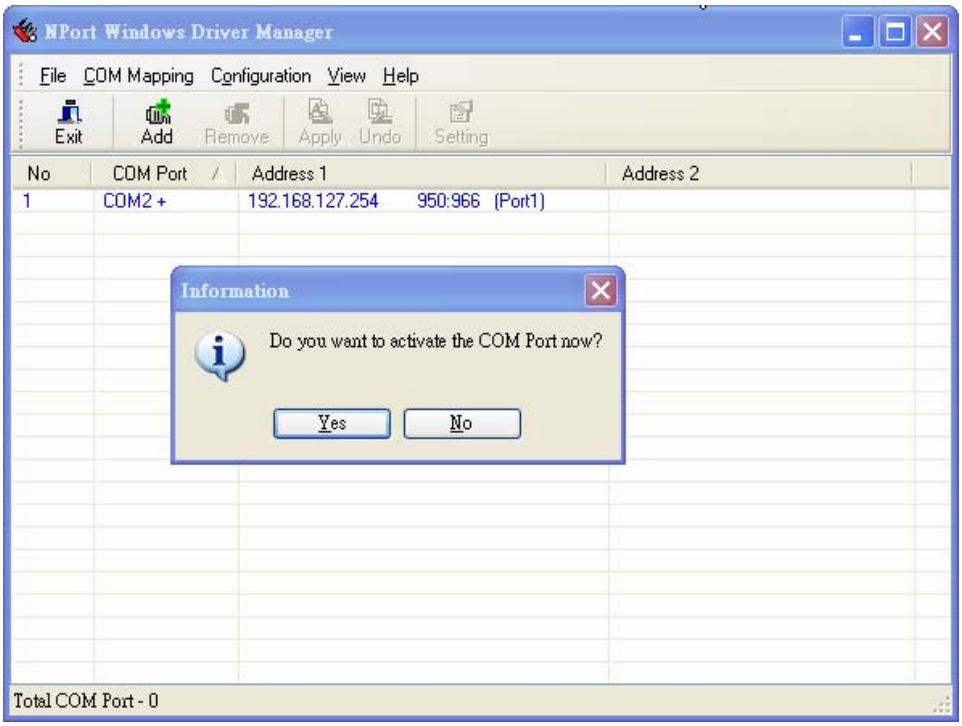
3. Click **Search** for MiiNePort E1 modules. From the list that is generated, select the server to which you will map COM ports, and then click **OK**.



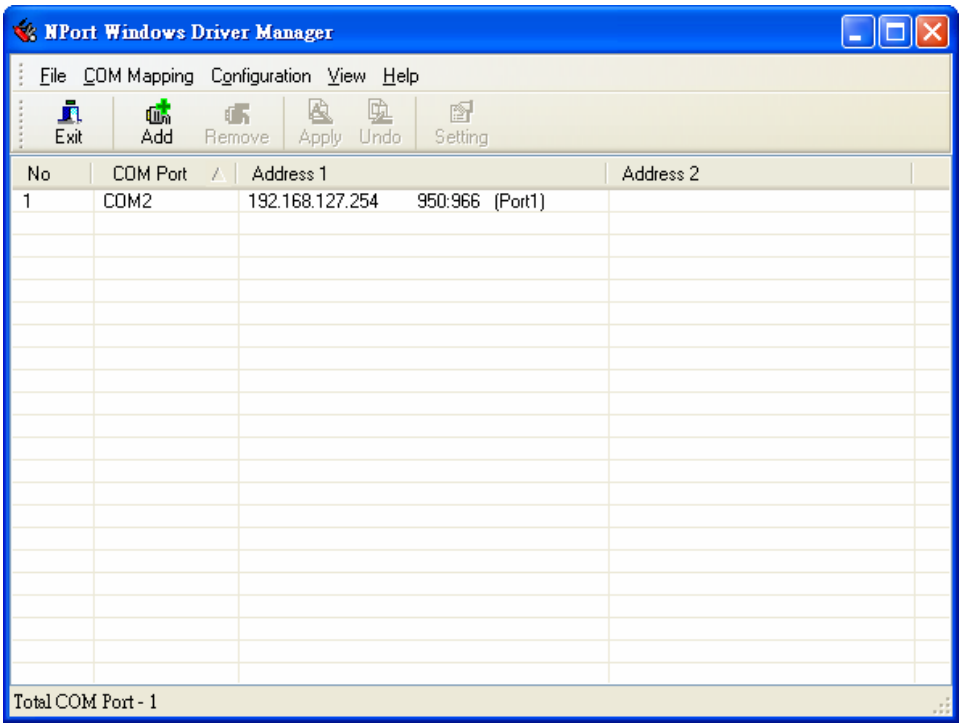
Alternatively, you can select **Input Manually** and then manually enter the MiiNePort E1 module's IP Address, 1st Data Port, 1st Command Port, and Total Ports to which COM ports will be mapped. Click **OK** to proceed to the next step. Note that the Add NPort page supports FQDN (Fully Qualified Domain Name), in which case the IP address will be filled in automatically.



4. COM ports and their mappings will appear in blue until they are activated. Activating the COM ports saves the information in the host system registry and makes the COM port available for use. The host computer will not have the ability to use the COM port until the COM ports are activated. Click **Yes** to activate the COM ports at this time, or click **No** to activate the COM ports later.


NPort Windows Driver Manager
File COM Mapping Configuration View Help
Exit Add Remove Apply Undo Setting
No COM Port Address 1 Address 2
1 COM2 + 192.168.127.254 950:966 (Port1)
Information
Do you want to activate the COM Port now?
Yes No
Total COM Port - 0

5. Ports that have been activated will appear in black.


NPort Windows Driver Manager
File COM Mapping Configuration View Help
Exit Add Remove Apply Undo Setting
No COM Port Address 1 Address 2
1 COM2 192.168.127.254 950:966 (Port1)
Total COM Port - 1

Installing Linux Real TTY Driver Files

1. Obtain the driver file from the included CD-ROM or the Moxa website, at http://www.moxa.com.
2. Log in to the console as a super user (root).
3. Execute **cd** / to go to the root directory.
4. Copy the driver file **npreal2xx.tgz** to the / directory.
5. Execute **tar xvfz npreal2xx.tgz** to extract all files into the system.
6. Execute **/tmp/moxa/mxinst.**

   For RedHat AS/ES/WS and Fedora Core1, append an extra argument as follows:

   # /tmp/moxa/mxinst SP1

   The shell script will install the driver files automatically.
7. After installing the driver, you will be able to see several files in the /usr/lib/npreal2/driver folder:

   > **mxaddsvr** (Add Server, mapping tty port)

   > **mxdelsvr** (Delete Server, un-mapping tty port)

   > **mxloadsvr** (Reload Server)

   > **mxmknod** (Create device node/tty port)

   > **mxrmnod** (Remove device node/tty port)

   > **mxuninst** (Remove tty port and driver files)

At this point, you will be ready to map the MiiNePort E1 serial port to the system tty port.

Mapping TTY Ports

Make sure that you set the operation mode of the serial port of MiiNePort E1 to Real COM mode. After logging in as a super user, enter the directory **/usr/lib/npreal2/driver** and then execute **mxaddsvr** to map the target MiiNePort E1 serial port to the host tty ports. The syntax of **mxaddsvr** is as follows:

mxaddsvr [MiiNePort E1 IP Address] [Total Ports] ([Data port] [Cmd port])

The **mxaddsvr** command performs the following actions:

1. Modifies **npreal2d.cf**.

2. Creates tty ports in directory /dev with major & minor number configured in **npreal2d.cf**.

3. Restarts the driver.

- **Mapping tty ports automatically**

  To map tty ports automatically, you may execute **mxaddsvr** with just the IP address and number of ports, as in the following example:

  # cd /usr/lib/npreal2/driver

  # ./mxaddsvr 192.168.3.4 16

  In this example, 16 tty ports will be added, all with IP 192.168.3.4, with data ports from 950 to 965 and command ports from 966 to 981.

- **Mapping tty ports manually**

  To map tty ports manually, you may execute **mxaddsvr** and manually specify the data and command ports, as in the following example:

  # cd /usr/lib/npreal2/driver

  # ./mxaddsvr 192.168.3.4 16 4001 966

  In this example, 16 tty ports will be added, all with IP 192.168.3.4, with data ports from 4001 to 4016 and command ports from 966 to 981.

Removing Mapped TTY Ports

After logging in as root, enter the directory **/usr/lib/npreal2/driver** and then execute **mxdelsvr** to delete a server. The syntax of mxdelsvr is:

mxdelsvr [IP Address]

Example:

 # cd /usr/lib/npreal2/driver

 # ./mxdelsvr 192.168.3.4

The following actions are performed when executing mxdelsvr:

1. Modify npreal2d.cf.

2. Remove the relevant tty ports in directory /dev.

3. Restart the driver.

If the IP address is not provided in the command line, the program will list the installed servers and total ports on the screen. You will need to choose a server from the list for deletion.

Removing Linux Driver Files

A utility is included that will remove all driver files, mapped tty ports, and unload the driver. To do this, you only need to enter the directory /usr/lib/npreal2/driver, then execute **mxuninst** to uninstall the driver. This program will perform the following actions:

1. Unload the driver.
2. Delete all files and directories in **/usr/lib/npreal2**
3. Delete directory **/usr/lib/npreal2**
4. Modify the system initializing script file.

7

Web Console Configuration

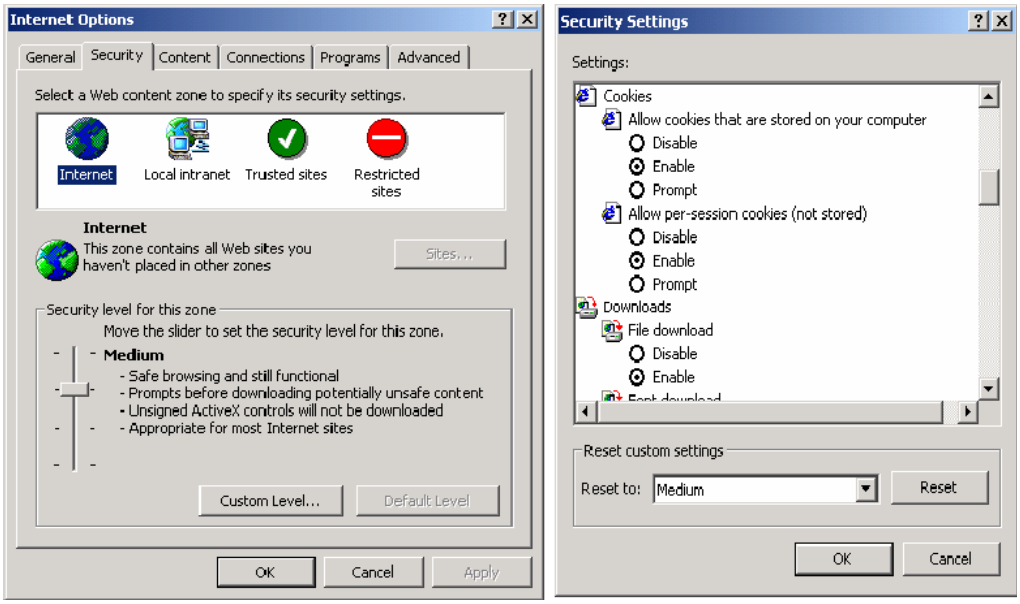
The web console is the most user-friendly way to configure your MiiNePort E1 Series module. This chapter introduces the web console function groups and function definitions.

This chapter includes the following sections:

- **Opening Your Brower**
- **Web Console Fundamentals**
- **Basic Settings**
  - Network Settings
  - Serial Port Settings
  - Operation Modes
- **Advanced Settings**
  - Accessible IP List
  - SNMP Agent
  - Pin and IO Settings
  - Serial Command Mode (SCM)
  - Miscellaneous
- **Maintenance**
  - Console Settings
  - Configuration Tool
  - Load Factory Default
  - Change Password

Opening Your Brower

1. Open your browser with the cookie function enabled. (To enable your Internet Explorer for cookies, right click on your desktop Internet Explorer icon, select Properties, click on the Security tab, and then select the three Enable options as shown in the figure below.)



2. Type 192.168.127.254 in the Address box (use the correct IP address if different from the default), and then press Enter.

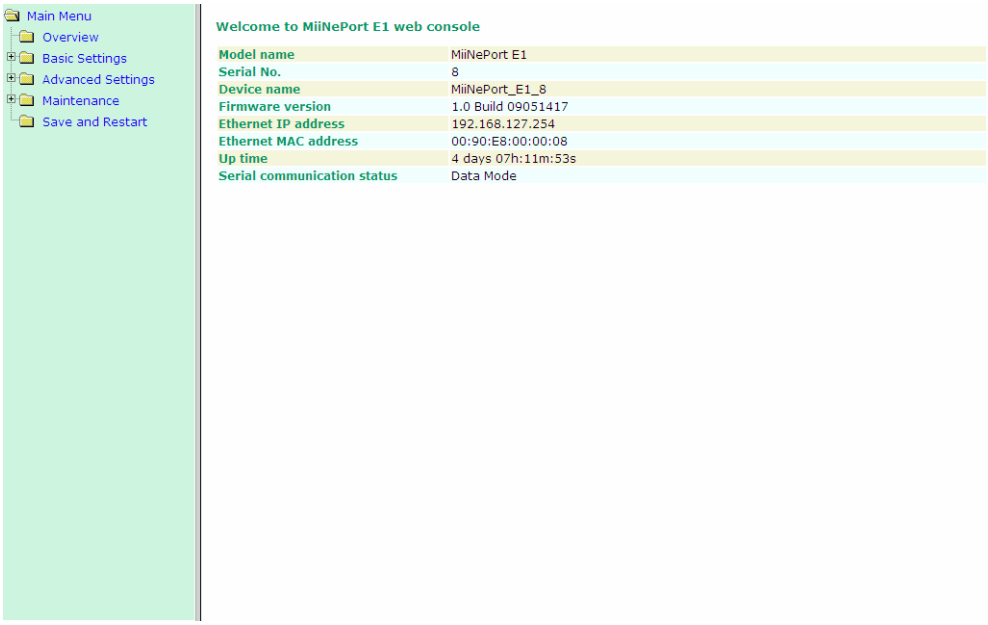
ATTENTION

If you use other web browsers, remember to enable the functions to "allow cookies that are stored on your computer" or "allow per-session cookies." MiiNePort E1 modules only use cookies for password transmission.

ATTENTION

Please refer to Chapter 5, Initial IP Address Configuration, for instructions on IP configuration. The examples in this chapter use the factory default IP address (192.168.127.254).

3. The web console will open. On this page, you can see a brief description of the web console's function groups in the left part of the page. You can also see a configuration overview of your MiiNePort E1 module.



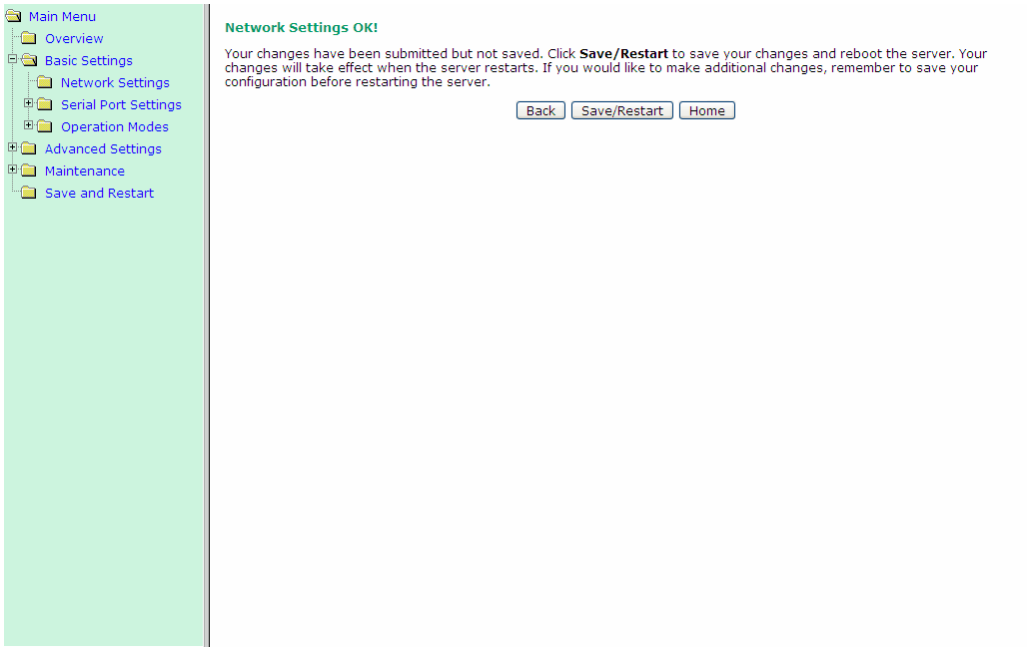
Web Console Fundamentals

On the web console, the left panel is the navigation panel and contains an expandable menu tree for navigating among the various settings and categories. When you click on a menu item in the navigation panel, the main window will display the corresponding options for that item.

Configuration changes can then be made in the main window. For example, if you click on **Basic Settings → Network Settings** in the navigation panel, the main window will show a page of network settings that you can configure.

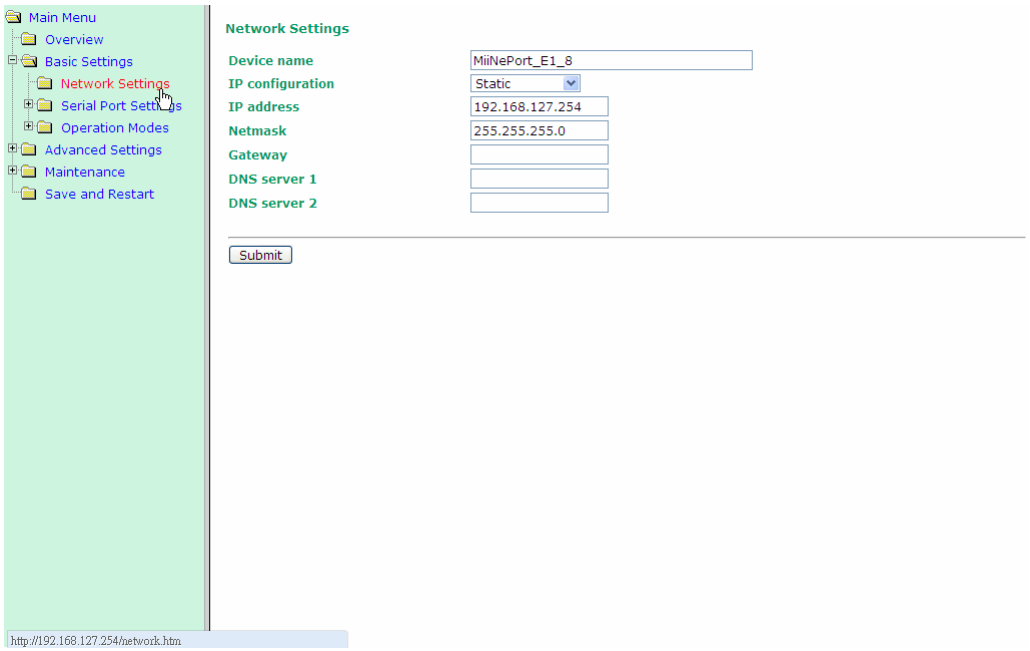
You must click on the **Submit** button to keep your configuration changes. The **Submit** button will be located at the bottom of every page that has configurable settings. If you navigate to another page without clicking the **Submit** button, your settings will not be retained.

Changes will not take effect until they are saved and the module is restarted! You may complete this in one step by clicking on **Save/Restart** after you submit a change. If you restart the module without saving your configuration, the module will discard all submitted changes.


Main Menu
Overview
Basic Settings
Network Settings
Serial Port Settings
Operation Modes
Advanced Settings
Maintenance
Save and Restart
Network Settings OK!
Your changes have been submitted but not saved. Click Save/Restart to save your changes and reboot the server. Your changes will take effect when the server restarts. If you would like to make additional changes, remember to save your configuration before restarting the server.
Back
Save/Restart
Home

Basic Settings

Network Settings



You must assign a valid IP address to the module before it will work in your network environment. Your network system administrator should provide you with an IP address and related settings for your network. The IP address must be unique within the network; otherwise the module will not have a valid connection to the network. First time users can refer to Chapter 5, Initial IP Address Configuration, for more information.

Device Name

Setting	Factory Default	Necessity
1 to 39 characters	NP [model name]_[Serial No.]	Optional

This option can be used to specify the location or application of the module, which may be useful when managing more than one module on the network.

IP configuration

Method	Function Definition
Static	User defined IP address, Netmask, Gateway.
DHCP	DHCP Server assigned IP address, Netmask, Gateway, DNS
DHCP/BOOTP	DHCP Server assigned IP address, Netmask, Gateway, DNS, or BOOTP Server assigned IP address
BOOTP	BOOTP Server assigned IP address
AUTOIP	AUTOIP protocols automatically negotiate and assign IP in 169.254/16 network

IP configuration is a required field. The default setting is Static.

ATTENTION

In Dynamic IP environments, the module will attempt to obtain an IP address from the DHCP or BOOTP server 3 times at 30-second intervals. The timeout for the first try will be 1 second, the second try will be 3 seconds, and the last try will be 5 seconds.

If the DHCP/BOOTP Server is unavailable, the module will use the default IP address (192.168.127.254), netmask, and gateway settings.

IP Address

Setting	Factory Default	Necessity
E.g., 192.168.1.1 (IP addresses of the form x.x.x.0 and x.x.x.255 are invalid.)	192.168.127.254	Required

An IP address is a number assigned to a network device, such as a computer, as a permanent address on the network. Computers use the IP address to identify and talk to each other over the network. Choose a proper IP address which is unique and valid in your network environment.

Netmask

Setting	Factory Default	Necessity
E.g., 255.255.255.0	255.255.255.0	Required

A subnet mask represents all the network hosts at one geographic location, in one building, or on the same local area network. When a packet is sent out over the network, the module will use the subnet mask to check whether the host specified in the packet is on a local network segment. If the address is on the same network segment as the module, a connection is established directly from the module. Otherwise, the connection is established through the default gateway.

Gateway

Setting	Factory Default	Necessity
E.g., 192.168.1.1	None	Optional

A gateway acts as an entrance to another network. Usually, the computers that control traffic within the network or at the local Internet service provider are gateway nodes. The module needs to know the IP address of the default gateway computer in order to communicate with the hosts outside the local network environment. For correct gateway IP address information, consult your network administrator.

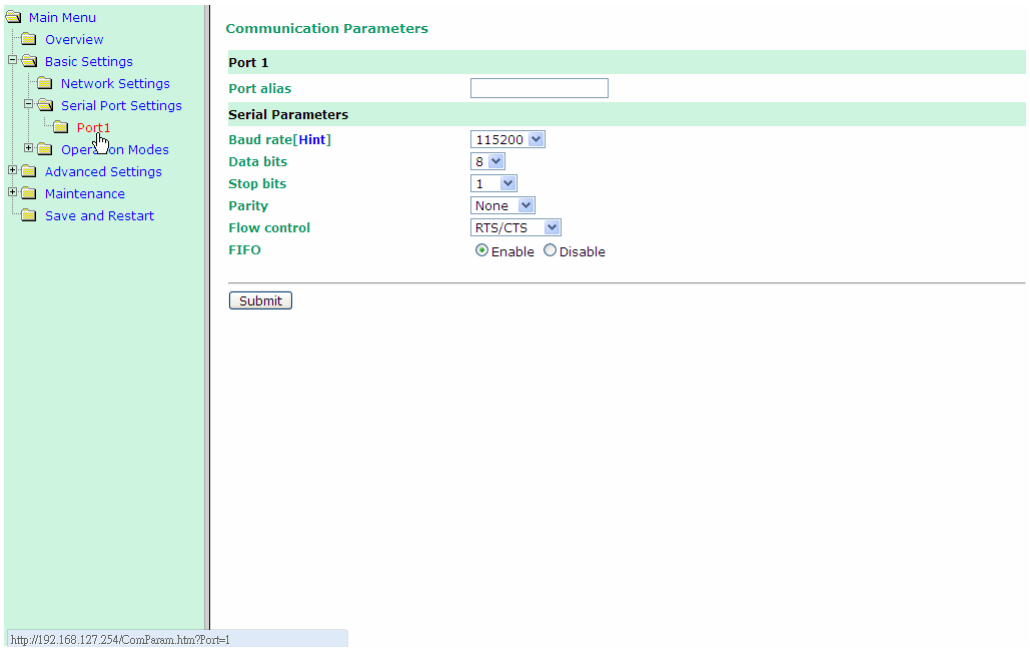
DNS server 1 / DNS server 2

Setting	Factory Default	Necessity
E.g., 192.168.1.1 (IP addresses of the form x.x.x.0 and x.x.x.255 are invalid)	None	Optional

Domain Name System (DNS) is how Internet domain names are identified and translated into IP addresses. A domain name is an alphanumeric name, such as moxa.com, that it is usually easier to remember. A DNS server is a host that translates the text-based domain name into the corresponding numeric IP address which is used to establish a TCP/IP connection. When the user enters a website address, the computer asks a DNS server for the website's IP address to connect to the web server.

When a DNS server is specified, the module acts as DNS client and will allow domain names instead of IP addresses to be used on the web console. The following web console fields support the use of domain names: TCP Client-Destination IP Address, and IP Address Report Server. Two DNS servers may be specified, DNS server 1 and DNS server 2. DNS server 2 is included for use when DNS sever 1 is unavailable.

Serial Port Settings



Port Alias

Setting	Factory Default	Necessity
1 to 15 characters (E.g., PLC-No.1)	None	Optional

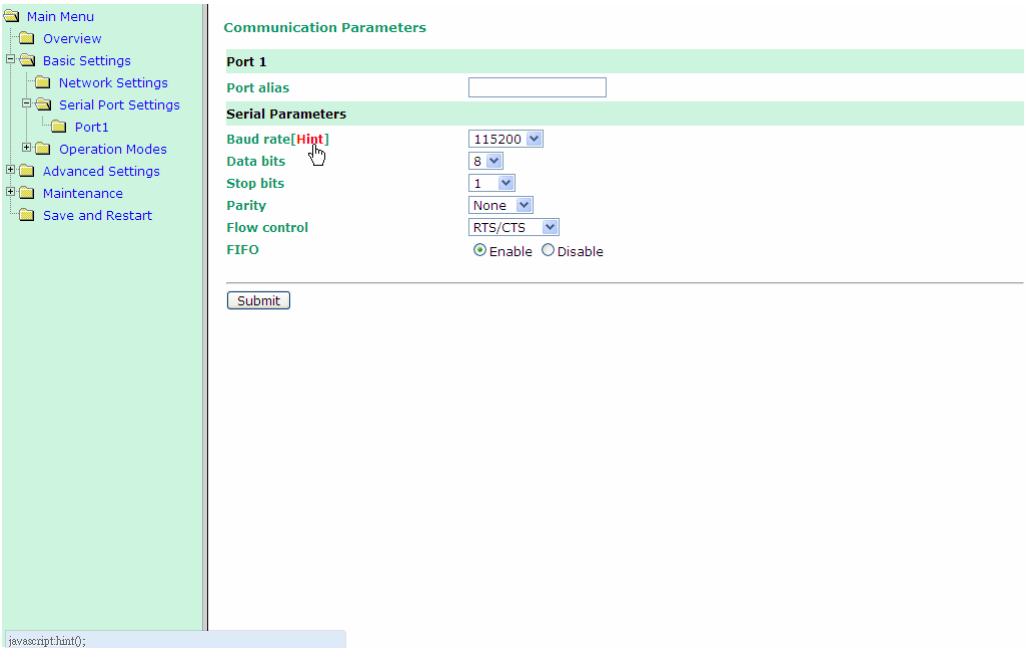
This function is designed for future use. You may enter a string to help in the module's serial port from other serial ports.

ATTENTION

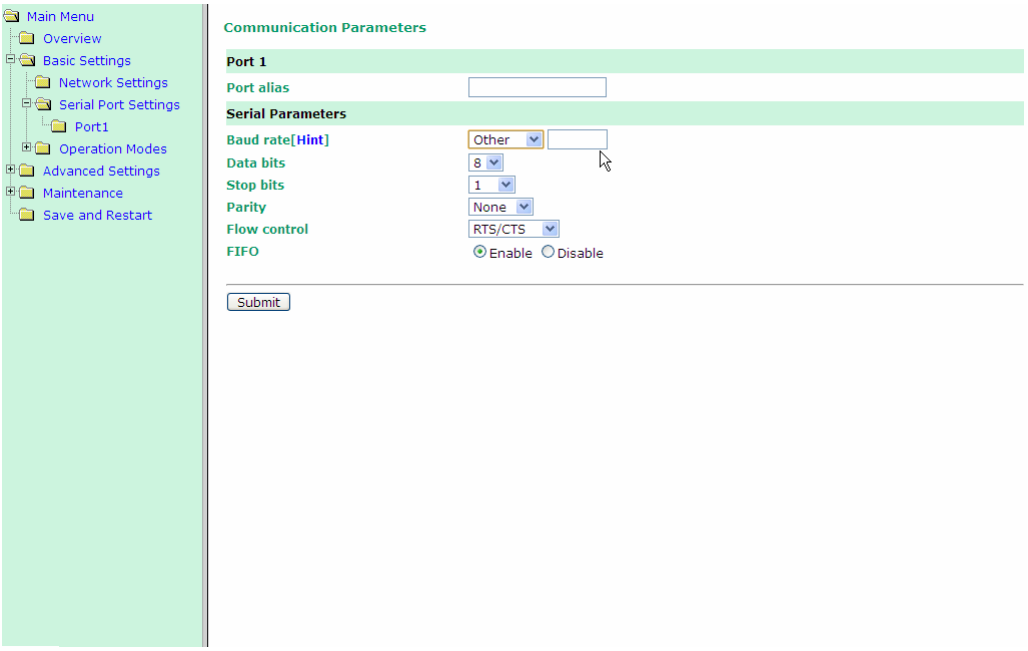
Refer to the serial communication parameters in your serial device's user's manual. The module's serial parameters should be the same as the parameters used by your serial device.

Baud Rate

Setting	Factory Default	Necessity
50 bps to 230.4 Kbps[7] (supports non-standard baudrates)	115.2 Kbps	Required



MiiNePort E1 supports the **Any Baud Rate (non-standard baud rate)** feature. If your baud rate is not listed, select **Other** from the drop-down list and type the baud rate in the input box. MiiNePort E1 will use the closest baud rate we can support.



[7] Baudrates up to 921.6 Kbps available by request on customized model.

Data Bits

Setting	Factory Default	Necessity
5, 6, 7, 8	8	Required

Stop Bits

Setting	Factory Default	Necessity
1, 2	1	Required

Stop Bits will be set to 1.5 when Data Bits is set to 5 bits.

Parity

Setting	Factory Default	Necessity
None, Even, Odd, Space, Mark	None	Required

Flow control

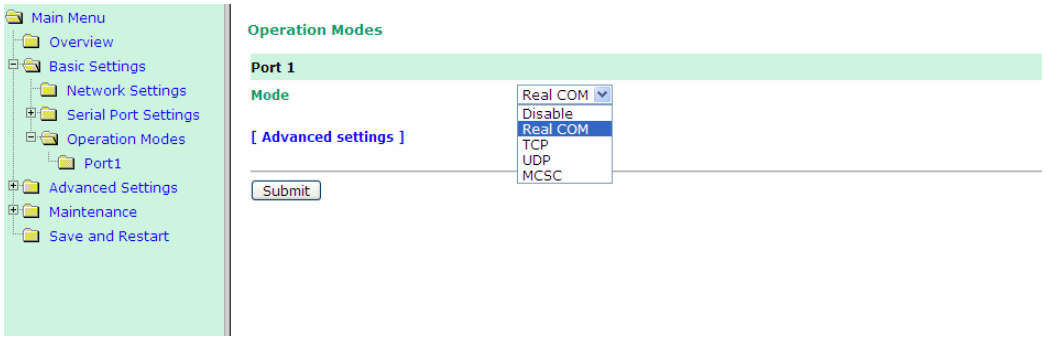
Setting	Factory Default	Necessity
None, RTS/CTS, DTR/DSR, XON/XOFF	RTS/CTS	Required

FIFO

Setting	Factory Default	Necessity
Enable, Disable	Enable	Required

Each module's serial port provides a 16-byte FIFO both in the Tx and Rx directions. Disable the FIFO setting when your serial device does not have a FIFO to prevent data loss during communication.

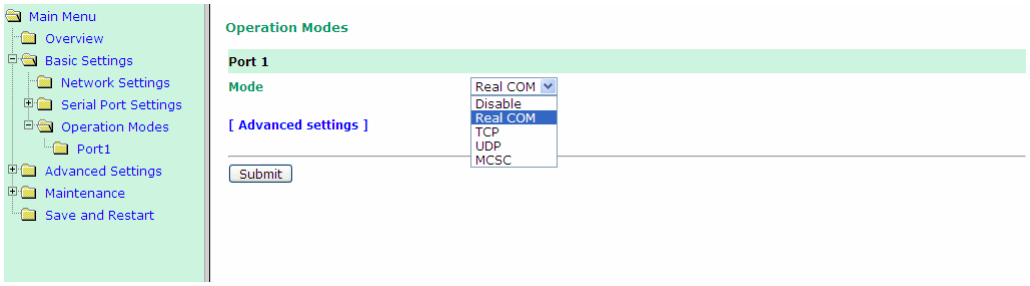
Operation Modes



Before get into this section, please refer to Chapter 3, Choosing the Proper Operation Mode, select a proper operation mode that best fits your device application.

To save your time, in each Operation Mode configuration page we only lay out the settings which we regard as necessary and mostly used. You can check and configure other functions by clicking **Advanced Settings** in each operation mode configuration page, however, we believe more than likely your application can work properly without making any changes in **Advanced Settings**.

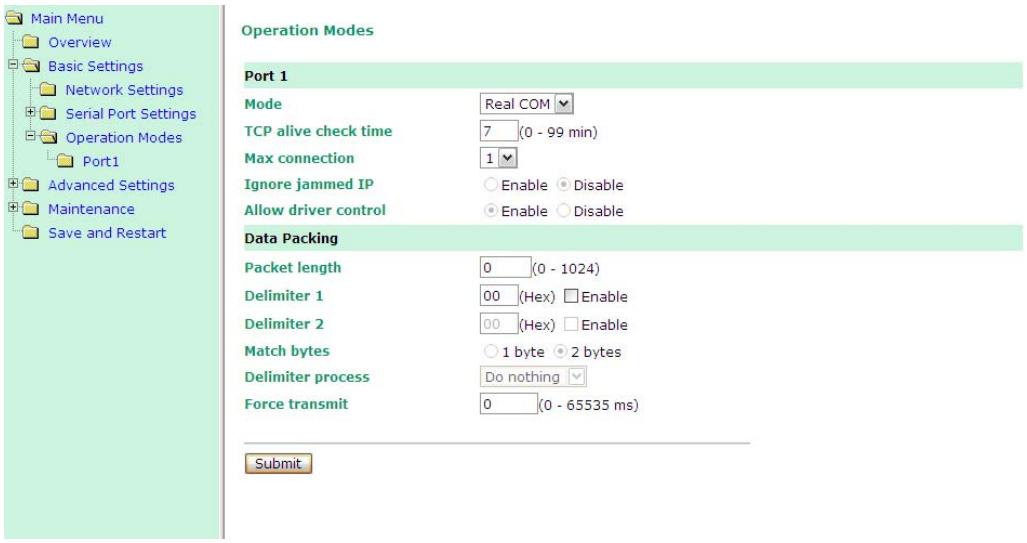
Real COM Mode



ATTENTION

To adopt Real COM mode, please refer to Chapter 6, Utility Console and Driver Installation, to install the Real COM driver on Windows or Linux.

Advanced Settings



TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

0 min: The TCP connection is not closed due to an idle TCP connection.

1 to 99 min: The module automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

Max connection

Setting	Factory Default	Necessity
1, 2, 3, 4	1	Required

Max connection is used when the device needs to receive data from different hosts simultaneously.

The factory default only allows 1 connection at a time. When Max Connection is set to 1, the Real COM driver on the specific host has full control.

Max connection 1: The module will only allow 1 host's Real COM driver to open a connection to the module's serial port.

Max connection 2 to 4: When set to 2 or higher, Real COM drivers for up to the specified number of hosts may open this port at the same time. When Real COM drivers for multiple hosts open the port at the same time, the COM driver only provides a pure data tunnel with no control ability. The serial port parameters will use firmware settings instead of your application program (AP) settings.

Application software that is based on the COM driver will receive a driver response of "success" when the software uses any of the Win32 API functions. The firmware will only send data back to the driver on the host. Data will be sent first-in-first-out when data is received by the MiiNePort E1 from the Ethernet interface.

ATTENTION

When **Max connection** is greater than 1, the MiiNePort E1 module will use a multiple connection application (i.e., 2 to 4 hosts are allowed access to the port at the same time). When using a multi connection application, the module will use the serial communication parameters as defined here in the web console, and all hosts connected to the port must use identical serial settings. If one of the hosts opens the COM port with different serial settings, data will not be transmitted properly.

Ignore jammed IP[9]

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required when Max connection greater than 1

This option determines how the port will proceed if multiple hosts are connected and one or more of the hosts stops responding as the port is transmitting data. If you select Disable, the port will wait until the data has been transmitted successfully to all hosts before transmitting the next group of data. If you select Enable, the port will ignore the host that stopped responding and continue data transmission to the other hosts.

[9] Ignore Jammed IP is only valid when the Max connection is greater than 1.

Allow driver control[10]

Setting	Factory Default	Necessity
Enable, Disable	Enable	Required when Max connection greater than 1

This option determines how the port will proceed if driver control commands are received from multiple hosts that are connected to the port. If Disable is selected, driver control commands will be ignored. If Enable is selected, control commands will be accepted, with the most recent command received taking precedence.

Packet length

Setting	Factory Default	Necessity
0 – 1024 bytes	0 byte	Required

The Packet length setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. At the default of 0 for packet length, no maximum amount is specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon as it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

The Delimiter fields are used to specify a character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character will be used, Delimiter 2 should be set to "0". If the delimiter will be a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to "0".

Note that data packing is not only controlled by the delimiter; it is also influenced by the module's buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. Also, if the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

[10] Allow drive control is only valid when the Max connection is greater than 1.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	1 byte (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receives either one of Delimiter 1 and Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receives both of Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	Do Nothing (only effective when Delimiter is enabled)	Optional

The Delimiter process field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

Do Nothing: Data in the buffer will be transmitted when the delimiter is received.

Delimiter + 1: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

Force transmit

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The **Force transmit** field is typically used in conjunction with the Delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data as soon as there is no serial communication for the specified time.

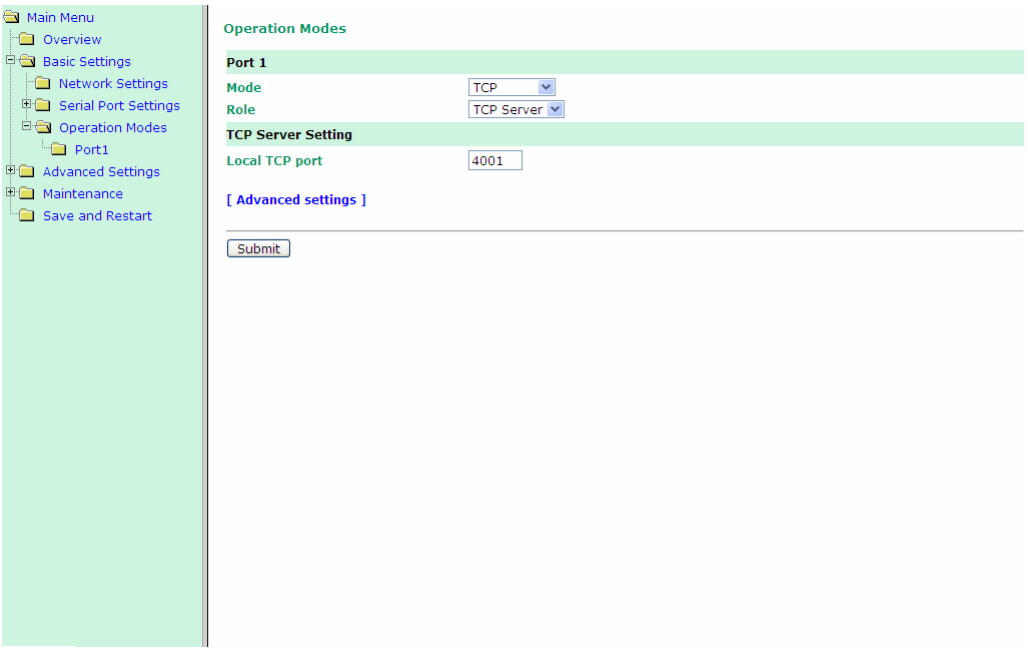
The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits needed to send one character is 10 bits, and the time required to transfer one character is

(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.

Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if no delimiter characters or Force transmit time were specified at all.

TCP Server Mode

To select TCP Server Mode, please select TCP as Mode first then select TCP Server as Role.

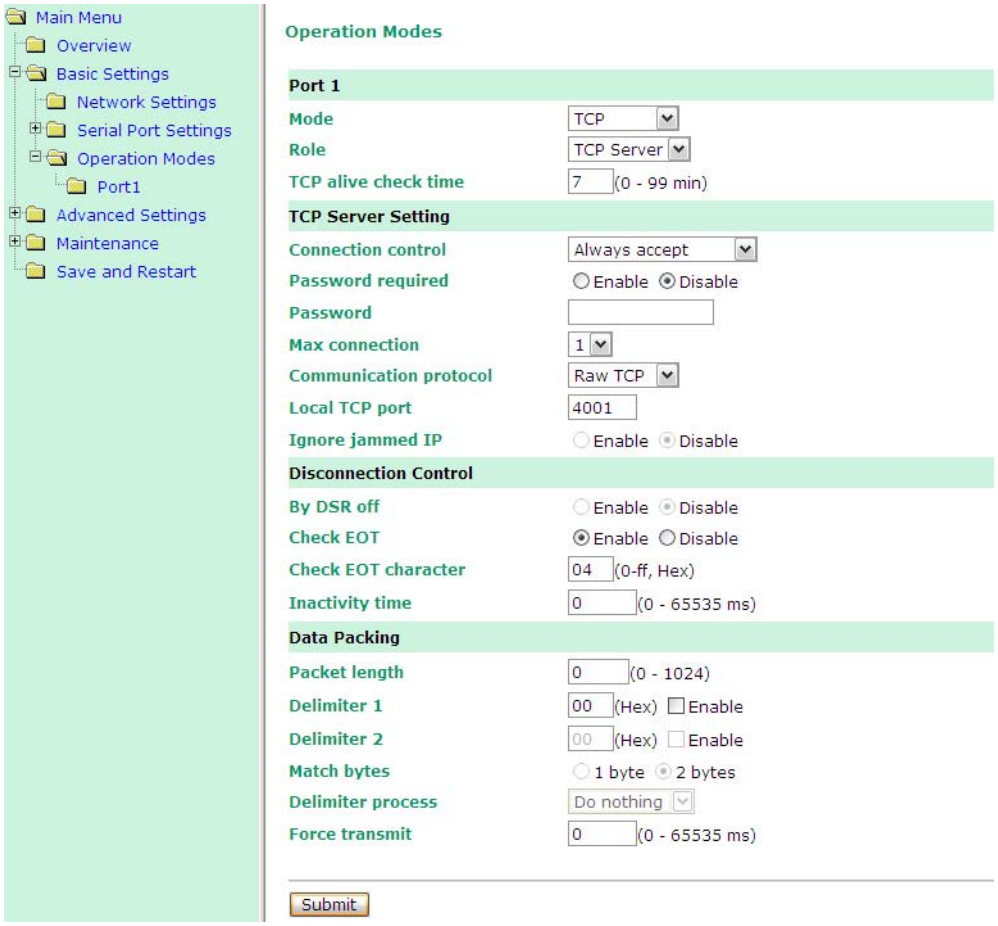


Local TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the TCP port that the module will use to listen to connections, and that other devices must use to contact the module. To avoid conflicts with well known TCP ports, the default is set to 4001.

Advanced Settings



TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

0 min: The TCP connection is not closed due to an idle TCP connection.

1 to 99 min: The module automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

Connection Control

Setting	Factory Default	Necessity
Always accept, Accept when DSR on	Always accept	Optional

Always accept: Always accept TCP connection from host.

Accept with DSR on: Accept TCP connection from host when the DSR is pulled **On**.

Default is **Always accept**.

ATTENTION

When setting the Connection Control or Disconnection Control by DSR signal of TCP Server mode, you must configure Pin 8 with the DSR function.

Password required

Setting	Factory Default	Necessity
Enable, Disable	Disable	Optional

Enable: When host is requesting a TCP connection to the MiiNePort E1 in TCP Server mode, the host will be required to enter a password.

Disable: The host can establish a TCP connection to the MiiNePort E1 without requiring a password.

Default is **Disable**.

Password

Setting	Factory Default	Necessity
1-15 English or number characters without space in between	None	Optional

After you enable the **Password required** function, you must set a set of passwords composed of 1 to 15 English or number characters without space in between.

Max connection

Setting	Factory Default	Necessity
1, 2, 3, 4	1	Required

Maximum connection is used when the device needs to receive data from different hosts simultaneously.

The factory default only allows 1 connection at a time. When Max Connection is set to 1, the Real COM driver on the specific host has full control.

Max connection 1: The module will only allow 1 host's Real COM driver to open a connection to the module's serial port.

Max connection 2 to 4: The module will allow requests from between 2 and 4 different hosts to open the module's serial port at the same time. When multiple hosts' Real COM drivers open the serial port at the same time, the COM driver only provides a pure data tunnel without control ability.

Application software that is based on the COM driver will receive a driver response of "success" when the software uses any of the Win32 API functions. The firmware will only send the data back to the driver on the host. Data will be sent first-in-first-out when data comes into the module from the Ethernet interface.

Communication Protocol

Setting	Factory Default	Necessity
Raw TCP, RFC 2217	Raw TCP	Required

RAW TCP: Standard TCP/IP protocol.

RFC 2217 [11]: Please refer Chapter 3, Choosing the Proper Operation Mode, for more details of RFC 2217 mode.

Default is **RAW TCP**.

Local TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the TCP port that the module will use to listen to connections, and that other devices must use to contact the module. To avoid conflicts with well known TCP ports, the default is set to 4001.

Ignore jammed IP[12]

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required when Max connection greater than 1

This option determines how the port will proceed if multiple hosts are connected and one or more of the hosts stop responding as the port is transmitting data. If you select Disable, the port will wait until the data has been transmitted successfully to all hosts before transmitting the next group of data. If you select Enable, the port will ignore the host that stopped responding and continue data transmission to the other hosts.

MiiNePort E1 can actively disconnect the connection established by the host.

By DSR off

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: MiiNePort E1 will actively cut off the connection when DSR is pulled **Off**.

Disable: Disable foresaid function

Default is **Disable**.

ATTENTION

When set the Connection Control or Disconnection Control by DSR signal, configure Pin 8 as DSR function is required.

[11] Be sure you install RFC 2217 driver before choosing RFC 2217

[12] Ignore Jammed IP is only valid when the Max connection is greater than 1.

Check EOT

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: Your device main system can send an serial EOT character to MiiNePort E1 to stop current TCP connection.

Disable: Disable this function.

Default is **Disable**.

Check EOT character

Setting	Factory Default	Necessity
0-ff, Hex	04	Optional

Set up EOT character.

Inactivity time

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0 ms: The TCP connection is not closed due to an idle serial line.

1-65535 ms: The module automatically closes the TCP connection if there is no serial data activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

This parameter defines the maintenance status as Closed or Listen on the TCP connection. The connection is closed if there is no incoming or outgoing data through the serial port during the specific Inactivity time.

If Inactivity time is set to 0, the current TCP connection is maintained until there is a connection close request. Even with Inactivity time is disabled, the module will still check the connection status between itself and the remote host by sending "keep alive" packets periodically. If the remote host does not respond to the packet, it assumes that the connection was closed down unintentionally. The module will then force the existing TCP connection to close.

ATTENTION

The Inactivity time should be longer than the Force transmit timeout. To prevent the unintended loss of data due a session getting disconnected, it is strongly recommended that this value is set large enough so that the intended data transfer is completed.

Packet length

Setting	Factory Default	Necessity
0 – 1024 bytes	0 byte	Required

The Packet length setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. At the default of 0 for packet length, no maximum amount is specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

The Delimiter fields are used to specify a character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character will be used, Delimiter 2 should be set to "0". If the delimiter will be a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to "0".

Note that data packing is not only controlled by the delimiter; it is also influenced by the module's buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. The module will also pack data for network transmissions if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	1 byte (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receives either one of Delimiter 1 and Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receives both of Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	Do Nothing (only effective when Delimiter is enabled)	Optional

The Delimiter process field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

Do Nothing: Data in the buffer will be transmitted when the delimiter is received.

Delimiter + 1: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

Force transmit

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The Force transmit field is typically used in conjunction with the Delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data as soon as there has been no serial communication for the specified time.

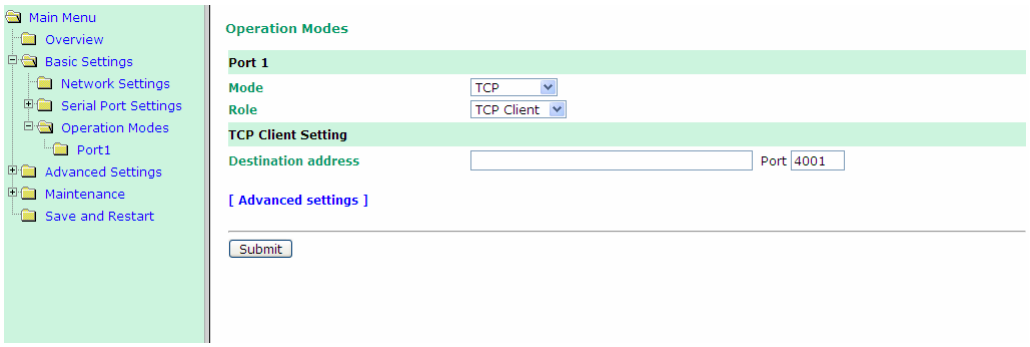
The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits needed to send one character is 10 bits, and the time required to transfer one character is

(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.

Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if no delimiter characters or Force transmit time were specified at all.

TCP Client Mode

To select TCP Client Mode, please select TCP as Mode first then select TCP Client as Role.



Destination address

Setting	Factory Default	Necessity
IP address or domain name (e.g., 192.168.1.1)	None	Required

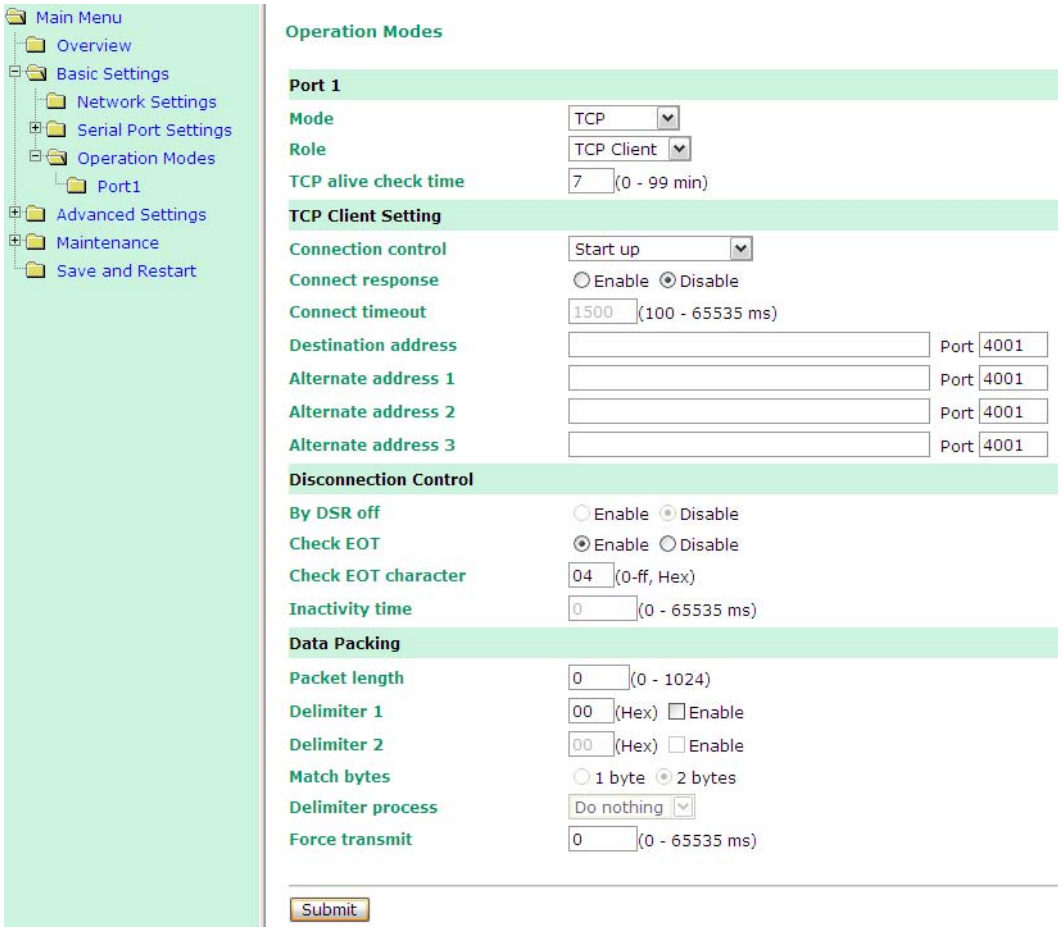
Use this field to specify the remote hosts that the module will connect to.

TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the TCP port of the destination host that will be connected to by MiiNePort E1. To avoid conflicts with well known TCP ports, the default is set to 4001.

Advanced Settings



TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

0 min: The TCP connection is not closed due to an idle TCP connection.

1 to 99 min: The module automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

Connection control

Setting	Factory Default	Necessity
Any character, Manual connection, Start up, With DSR on	Any character	Optional

Any character: The module will attempt to establish a TCP connection as soon as the module starts receiving serial data from your device's main system.

Manual connection: Your device main system can request MiiNePort E1 establish a connection by command. The command should begin with "C" and end with "LF" or "CR-LF", and IP address/port in the middle. For example: C192.168.32.221/4001.

Start up: The module will attempt to establish a TCP connection as soon as it is powered on.

With DSR on: The module will attempt to establish a TCP connection with the DSR being pulled **On**.

ATTENTION

When setting the Connection Control or Disconnection Control by DSR signal of TCP Client mode, you must configure Pin 8 with the DSR function.

Connect response

Setting	Factory Default	Necessity
Enable, Disable	Disable	Optional

Enable: MiiNePort E1 will send a one-byte message back to device main system reporting the connection status; "N" means connection failed to be established. "C" means connection successfully established. "D" means the connection lost.

Disable: Disable above function.

Default is **Disable**.

Connect Timeout [13]

Setting	Factory Default	Necessity
100 - 65535 ms	1500 ms	Required

This is to limit the time MiiNePort E1, as a TCP Client, try to establish a connection to the Destination address. When the time reaches, MiiNePort E1 will abort the attempt.

MiiNePort E1 can actively disconnect the connection established by the host.

Destination address 1/2/3/4 [14]

Setting	Factory Default	Necessity
IP address or domain name (e.g., 192.168.1.1)	None	Required

Use this field to specify the remote hosts that the module will connect to.

[13] Connection timeout is only valid with manual connections.

[14] If you already enter one set of Destination address and Port, you will see that the data is saved when you enter Advanced settings.

ATTENTION

Up to 4 connections can be established between the module and hosts. The connection speed or throughput may be low if one of the four connections is slow. In this case, the other 3 connections will be delayed while waiting for the slowest connection to finish transmitting or receiving.

ATTENTION

The **Destination address** parameter can accept domain name addresses in place of IP addresses if a DNS server has been configured. For some applications, you may need to send the data actively to a remote destination's domain name address.

By DSR off

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: MiiNePort E1 will actively cut off the connection when DSR is pulled **Off**.

Disable: Disable foresaid function.

Default is **Disable**.

Check EOT

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: Your device main system can send an EOT character to the MiiNePort E1 to stop current TCP connection.

Disable: Disable this function.

Default is **Disable**.

Check EOT character

Setting	Factory Default	Necessity
0-ff, Hex	04	Optional

Set up EOT character.

Inactivity time

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0 ms: The TCP connection is not closed due to an idle serial line.

1-65535 ms: The module automatically closes the TCP connection if there is no serial data activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

This parameter defines the maintenance status as Closed or Listen on the TCP connection. The connection is closed if there is no incoming or outgoing data through the serial port during the specific Inactivity time.

If Inactivity time is set to 0, the current TCP connection is maintained until there is a connection close request. Even with Inactivity time is disabled, the module will still check the connection status between itself and the remote host by sending "keep alive" packets periodically. If the remote host does not respond to the packet, it assumes that the connection was closed down unintentionally. The module will then force the existing TCP connection to close.

ATTENTION

The **Inactivity time** should be longer than the **Force transmit** timeout. To prevent the unintended loss of data due a session getting disconnected, it is strongly recommended that this value is set large enough so that the intended data transfer is completed.

Packet length

Setting	Factory Default	Necessity
0 – 1024 bytes	0 byte	Required

The Packet length setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. At the default of 0 for packet length, no maximum amount is specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

The Delimiter fields are used to specify a character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character will be used, Delimiter 2 should be set to "0". If the delimiter will be a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to "0".

Note that data packing is not only controlled by the delimiter; it is also influenced by the module's buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. Also, if the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	1 byte (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receive either one of Delimiter 1 and Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receive both of Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	Do Nothing (only effective when Delimiter is enabled)	Optional

The Delimiter process field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

Do Nothing: Data in the buffer will be transmitted when the delimiter is received.

Delimiter + 1: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

Force transmit

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will packed the data in its buffer into the same data frame for network transmission.

The Force transmit field is typically used in conjunction with the Delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data as soon as there has been no serial communication for the specified time.

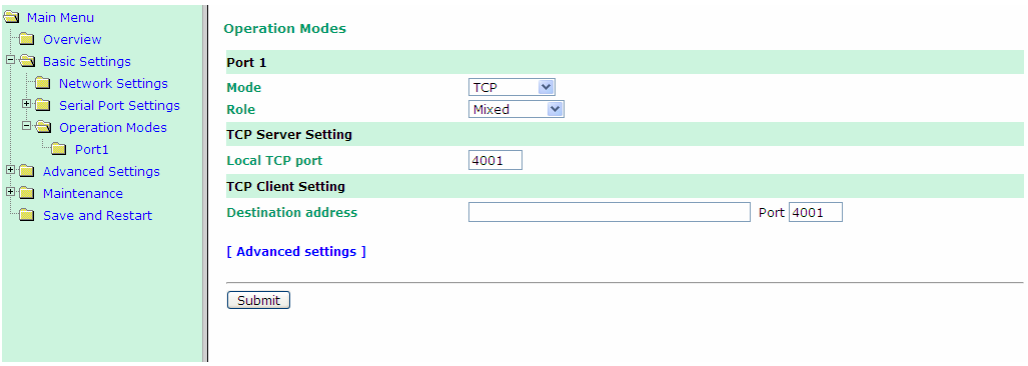
The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits needed to send one character is 10 bits, and the time required to transfer one character is

(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.

Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if no delimiter characters or Force transmit time were specified at all.

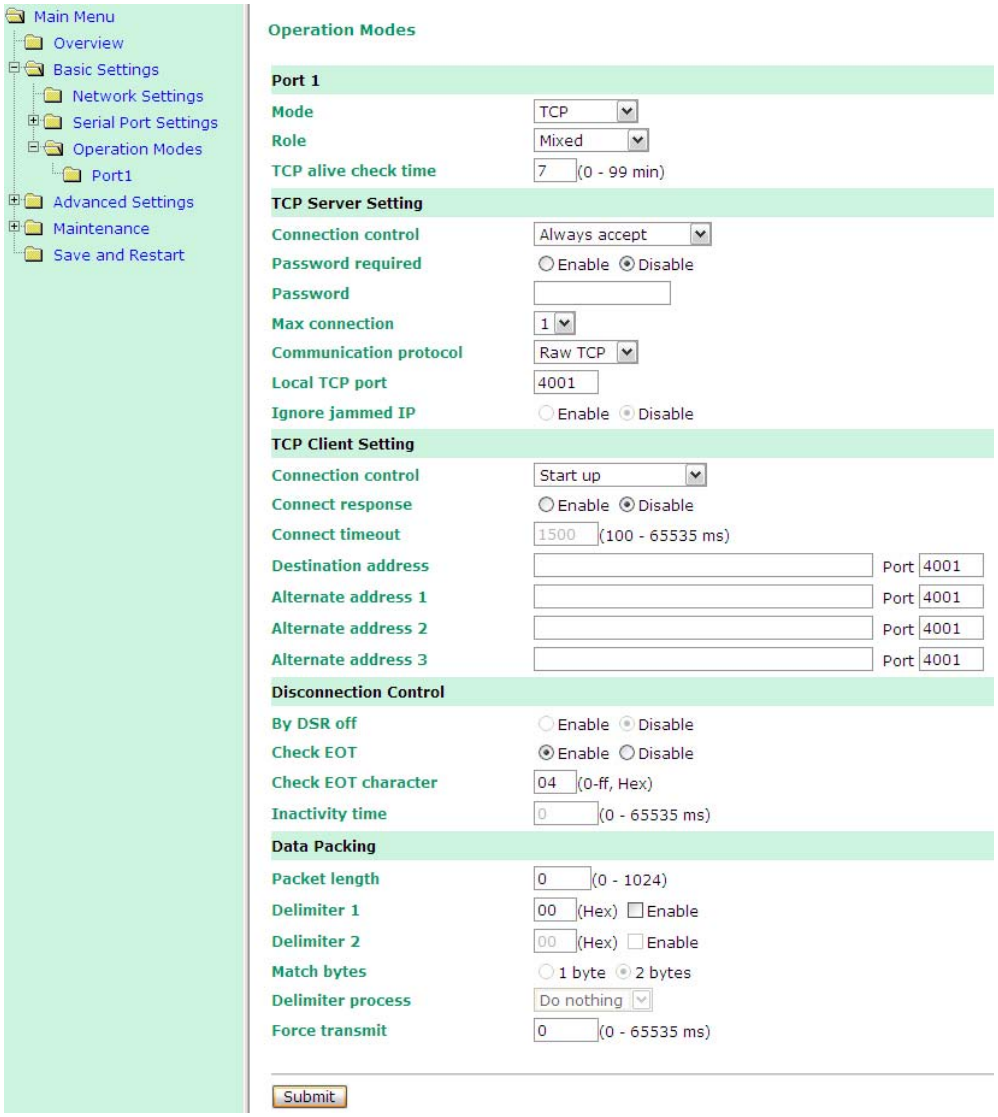
TCP Mixed Mode

To select TCP Mixed Mode, please select TCP as Mode first then select TCP Mixed as Role.

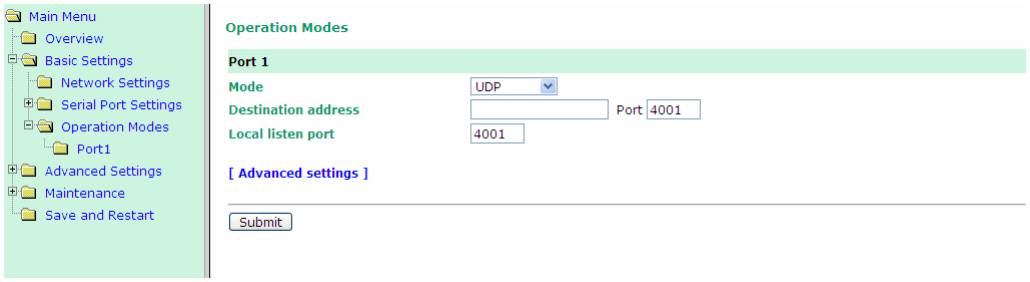


For all configuration settings in **TCP Mixed Mode**, please refer to the settings describe in **TCP Server Mode** and **TCP Client Mode**.

Advanced Settings



UDP Mode



Destination address

Setting	Factory Default	Necessity
IP address or domain name (e.g., 192.168.1.1)	None	Required

Use this field to specify the remote hosts that the module will connect to.

Port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

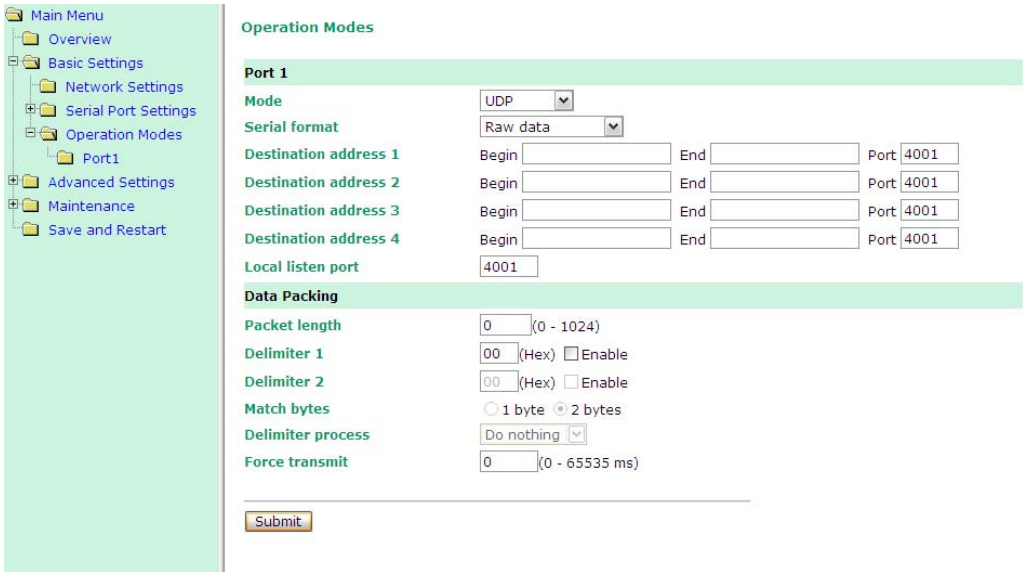
Use this field to indicate the UDP port of the destination host that will be connected to by MiiNePort E1. To avoid conflicts with well known UDP ports, the default is set to 4001.

Local listen port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the local listen UDP port of the MiiNePort E1. To avoid conflicts with well known UDP ports, the default is set to 4001.

Advanced Settings



Serial format

Setting	Factory Default	Necessity
Raw data, Packet command	Raw data	Optional

When the MiiNePort E1 receives serial data and then sends out a UDP packet, it is possible to dynamically change the destination address during this operation with the **Serial Format** function. If you set the Serial format as **Raw data**, you can then set the up to four desired destinations. But if you set it as **Packet command**, you will then be able to use the following command to change the destination IP address on the fly.

The command format is:

Start byte + Destination IP + Length + Data

For example, if you want to change the destination IP address to 192.168.35.100 with data "0123", you will need to send the serial data as 02C0A82364000430313233.

The ASCII to Hex code transformation of this example is shown below:

	Start byte	Destination IP				Length (2 bytes)	Data
ASCII	SOT	192	168	35	100	04	0123
Hex	02	C0	A8	23	64	00 04	30 31 32 33

Please also note that the destination port should be the same as the local listen port.

Destination address 1/2/3/4 [16]

Setting	Factory Default	Necessity
IP address range (e.g., Begin: 192.168.1.1 End: 192.168.1.10) or single IP address (e.g., Begin 192.168.1.1)	None	Required

Use this field to specify the remote hosts that the module will connect to.

ATTENTION

The **Destination address** parameter can accept domain name addresses in place of IP addresses if a DNS server has been configured. For some applications, you may need to send the data actively to a remote destination's domain name address.

ATTENTION

The maximum selectable IP address range is 64 addresses. However, when using multi-unicast, you may enter IP addresses of the form xxx.xxx.xxx.255 in the Begin field. For example, enter 192.127.168.255 to allow the MiiNePort E1 to broadcast UDP packets to all hosts with IP addresses between 192.127.168.1 and 192.127.168.254.

[16] If you already enter one set of **Destination address** and **Port**, you will see the data is kept when you enter into **Advanced settings**.

Packet length

Setting	Factory Default	Necessity
0 – 1024 bytes	0 byte	Required

The **Packet length** setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. At the default of 0 for packet length, no maximum amount is specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	"0" for None	Optional

The Delimiter fields are used to specify a character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character is used, Delimiter 2 should be set to "0". If the delimiter is a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to "0".

Note that data packing is not only controlled by the delimiter; it is also influenced by the module's buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. Also, if the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	1 byte (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receives either Delimiter 1 or Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receives both Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	Do Nothing (only effective when Delimiter is enabled)	Optional

The Delimiter process field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

Do Nothing: Data in the buffer will be transmitted when the delimiter is received.

Delimiter + 1: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

Force transmit

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The Force transmit field is typically used in conjunction with the Delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

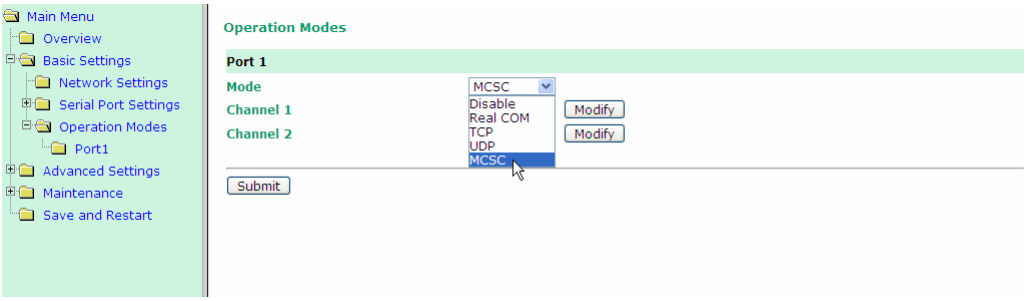
When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data as soon as there has been no serial communication for the specified time.

The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits needed to send one character is 10 bits, and the time required to transfer one character is

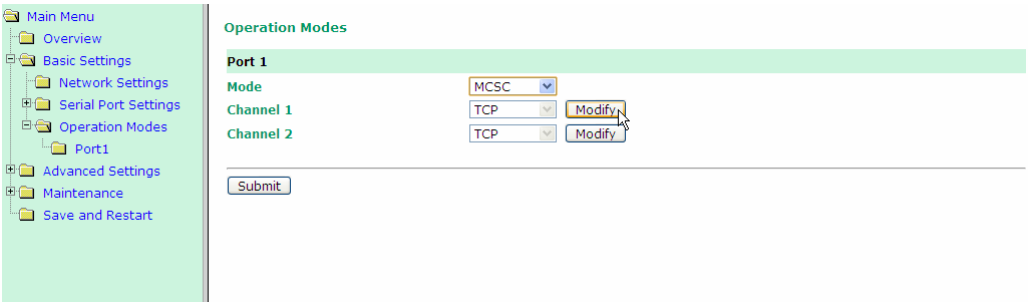
(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.

Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if no delimiter characters or Force transmit time were specified at all.

MCSC Mode



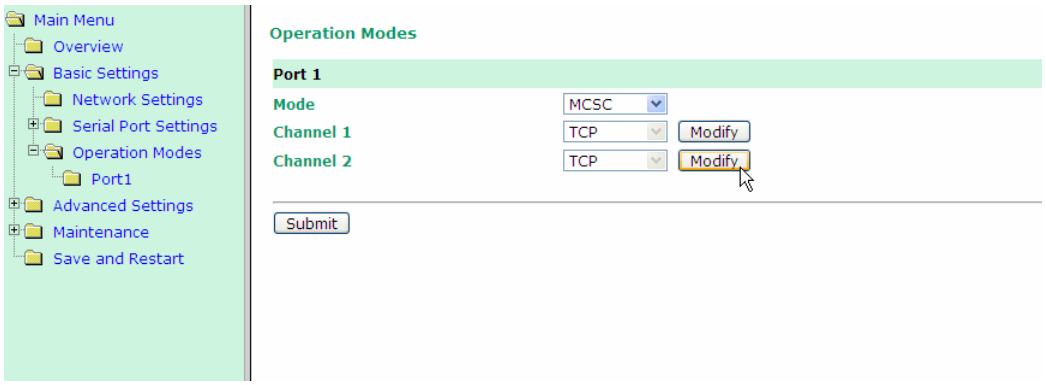
After the **MCSC** mode is selected, you will see 2 channels that are ready for your configuration. Under the **MCSC** structure, each channel works independently so you need to configure each channel separately according to your application. Click on **Modify** to configure the **Channel 1**.



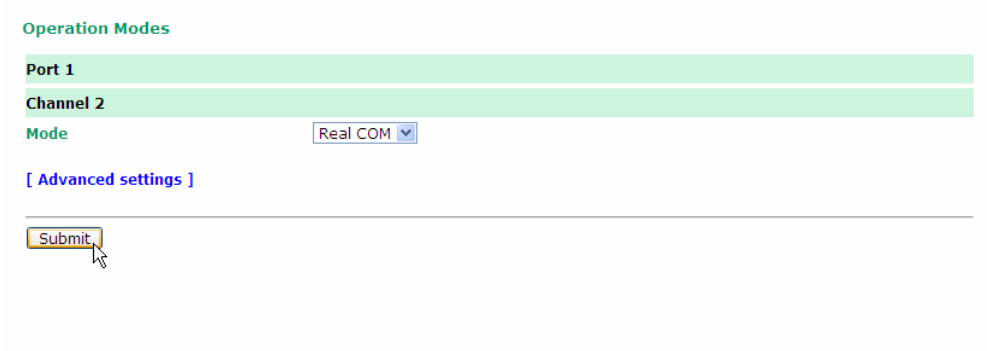
After you click **Modify,** the channel configuration window will pop up and you will see exactly the same configuration options as stated earlier in this chapter. For example, if you want to set **Channel 1** to **TCP Client** mode, just finish your configuration, click **Submit**, and a confirmation window will appear.

Operation Modes Settings OK!
Your changes have been submitted but not saved. Click Save/Restart to save your changes and reboot the server. Your changes will take effect when the server restarts. If you would like to make additional changes, remember to save your configuration before restarting the server.
Back
Save/Restart
Close

We suggest that you to click **Close** to go back to the **Operation Modes** configuration screen to finish configuring your **Channel 2**. After you get back to the screen, you will see the **TCP** mode setting for **Channel 1** is already shown on the screen. Please note that after you click **Submit**, your configuration is already stored in MiiNePort E1's memory and won't be erase unless your change the configuration again.

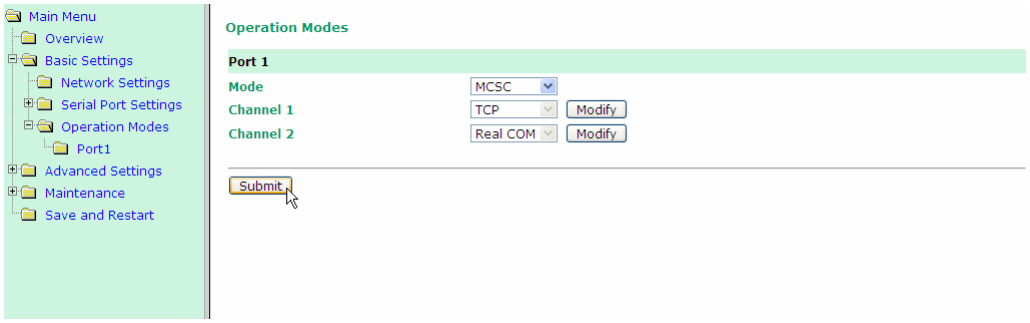


Click **Modify** to configure Channel 2. For example, if you want to set **Channel 2** to **Real COM** mode, you need to finish your **Real COM** settings and then click **Submit**.

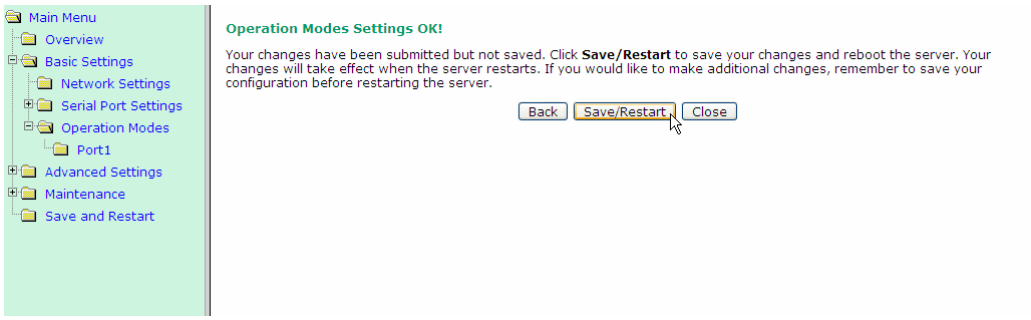


A confirmation window will then appear; click **Close** to return to the **Operation Modes** setting screen.

Click **Submit** to finish **MCSC** configuration on your MiiNePort E1.



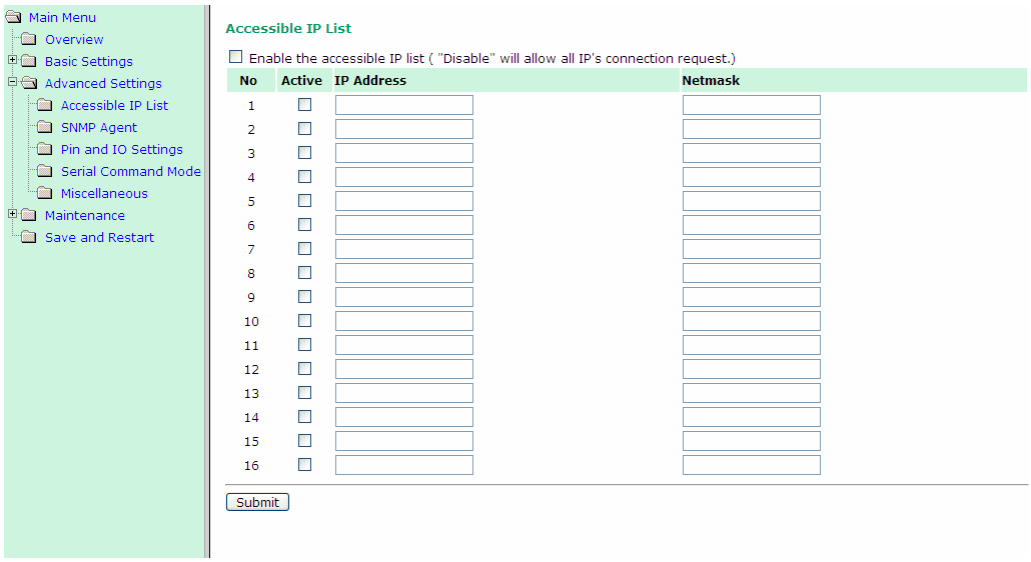
When the confirmation screen appears, choose either **Save/Restart** to activate the changes you've made or **Close** to continue making other configurations.


Main Menu
Overview
Basic Settings
Network Settings
Serial Port Settings
Operation Modes
Port1
Advanced Settings
Maintenance
Save and Restart
Operation Modes Settings OK!
Your changes have been submitted but not saved. Click Save/Restart to save your changes and reboot the server. Your changes will take effect when the server restarts. If you would like to make additional changes, remember to save your configuration before restarting the server.
Back
Save/Restart
Close

Besides the configuration settings, please refer to **Chapter 8, NetEZ Technologies, MCSC** for the **MCSC** command format and the device system design guidance.

Advanced Settings

Accessible IP List



- **To allow access to a specific IP address**

  Enter the IP address in the corresponding field; enter **255.255.255.255** for the netmask.

- **To allow access to hosts on a specific subnet**

  For both the IP address and netmask, use **0** for the last digit (e.g., **192.168.1.0** and **255.255.255.0**).

- **To allow unrestricted access**

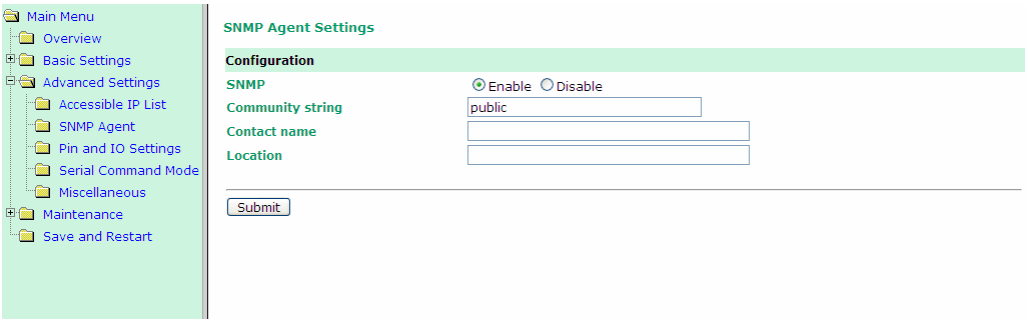
  Deselect the **Enable the accessible IP list** option.

Refer to the following table for more configuration examples.

Allowed Hosts	Entered IP address/Netmask
Any host	Disable
192.168.1.120	192.168.1.120 / 255.255.255.255
192.168.1.1 to 192.168.1.254	192.168.1.0 / 255.255.255.0
192.168.0.1 to 192.168.255.254	192.168.0.0 / 255.255.0.0
192.168.1.1 to 192.168.1.126	192.168.1.0 / 255.255.255.128
192.168.1.129 to 192.168.1.254	192.168.1.128 / 255.255.255.128

SNMP Agent

The enable the SNMP agent function, select the **Enable** option for SNMP under Configuration.



Community string

Setting	Factory Default	Necessity
1 to 39 characters (E.g., support, 886-89191230 #300)	Public	Optional

A community name is a plain-text password mechanism that is used to authenticate queries to agents of managed network devices.

Contact Name

Setting	Factory Default	Necessity
1 to 39 characters (E.g., support, 886-89191230 #300)	None	Optional

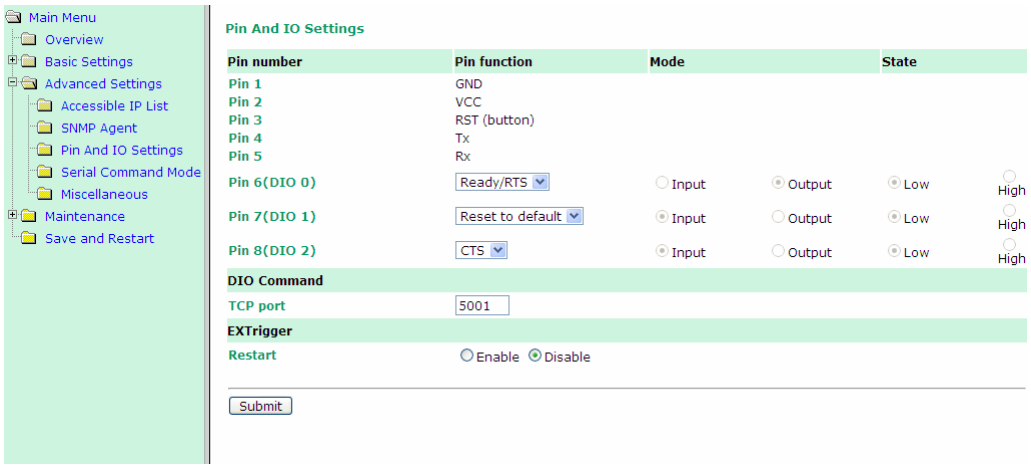
The SNMP contact information usually includes an emergency contact name and telephone or pager number.

Location

Setting	Factory Default	Necessity
1 to 39 characters (E.g., floor 1, office 2)	None	Optional

Enter a location string for SNMP agents. This string is usually set to the street address where the module is physically located.

Pin and IO Settings



Pin Function

For the 3 configurable pins (Pin 6, 7, and 8), refer to **Chapter 1, Pin Assignments** for their default settings [17] and change to the appropriate function for your application.

DI is for digital input operation, where the channel's status is controlled by an external digital switch. DO is for digital output operation, where the channel transmits a high or low signal. Use **Status** to control high or low status for digital output channels; status setting will be ignored for digital input channels.

Reset to Default (Use Pin 7, DIO 1) is used to clear the password or reset the MiiNePort E1 to Moxa's factory default settings. When this function is enabled, Pin 7, DIO 1, will be forced to digital input mode and will act as an internal reset mechanism. Pulling Pin 7, DIO 1, "low" for 100 ms will load the Moxa factory default settings. This function is enabled by default.

DIO Command

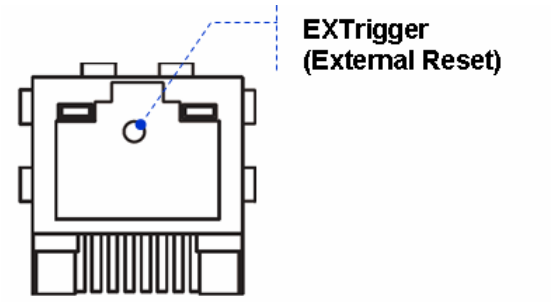
DIO Command TCP Port is the port number that will be reserved for DIO commands. DIO commands may be used to control and obtain data from the module's DIO channels. Please refer to Appendix D for additional information on DIO commands.

[17] Default settings: Pin 6 as Ready/RTS, Pin 7 as Reset to Default, Pin 8 as CTS.

EXTrigger

MiiNePort E1's **EXTrigger** has 2 functions; reset the module to Moxa's default configuration and restarting the Module. EXTrigger is designed to provide an easy tool for network troubleshooting without interrupting device operation.

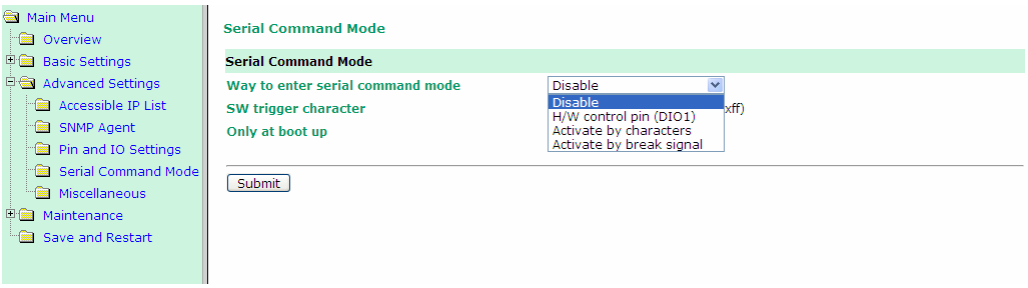
The **Reset to Default** function is enabled by default and is not subject to be disabled. You can choose to enable or disable the **Restart** function in the **Pin and IO Settings**.



For more details on how to take advantage of the EXTrigger, please refer to Chapter 8, NetEZ Technologies' EXTrigger (External Trigger).

Serial Command Mode (SCM)

SCM (Serial Command Mode) uses serial communication between the MiiNePort E1 and your device's main system to configure the MiiNePort E1, usually during device operation. For more details about SCM commands, please refer to Chapter 8, NetEZ Technologies.

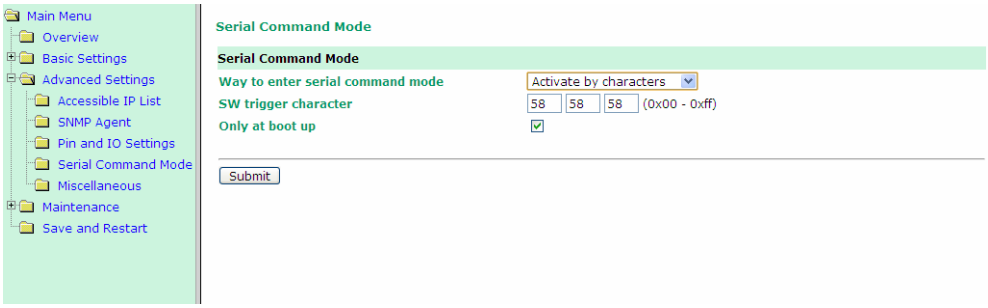


Way to enter serial command mode

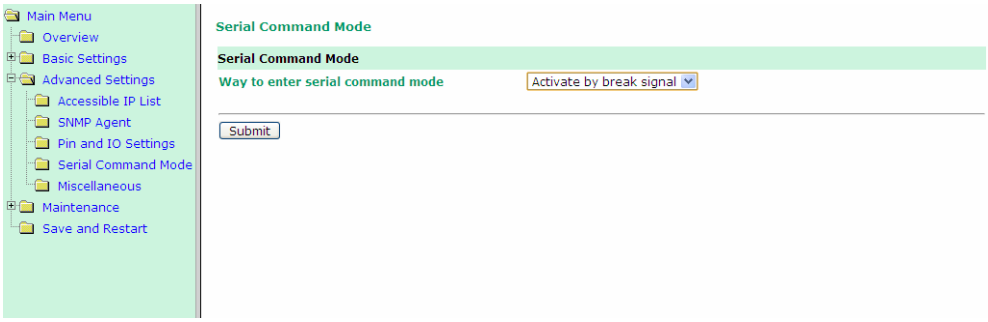
Setting	Factory Default	Necessity
Disable, H/W control pin (DIO1), Activate by characters, Activate by break signal	Activate by characters (Only at boot up)	Required

We offer 3 ways for you to access SCM:

- **H/W control pin (DIO 1):** Access SCM by pulling Pin 7, DIO1, to "low".
- **Activate by characters:** Access SCM by sending the MiiNePort E1's configurable three characters from your device's main system. Check **Only at boot up** if you want to make this effective only when booting up the MiiNePort E1.



- **Activate by break signal:** Break signals are caused by sending continuous Spacing values (no Start or Stop bits). When there is no electricity present on the data circuit, the line is considered to be sending a Break. The Break signal must be of duration longer than the time it takes to send a complete byte plus Start, Stop and Parity bits.

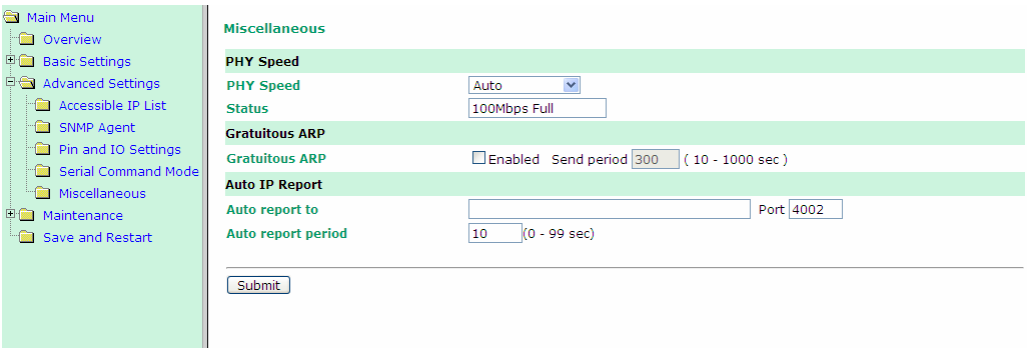


Miscellaneous

PHY Speed

Setting	Factory Default	Necessity
Auto, 100 Mbps Full Duplex, 100 Mbps Half Duplex, 10 Mbps. Full Duplex, and 10 Mbps Half Duplex.	Auto	Required

You can select the **PHY speed** from Auto, 100 Mbps Full Duplex, 100 Mbps Half Duplex, 10 Mbps Full Duplex, and 10 Mbps Half Duplex.



Auto IP Report

When the module is used in a dynamic IP environment, additional time must be spent on IP management tasks. For example, when a module is operating as a server (TCP or UDP modes), a PC operating as a client will need to know the module's IP address. If the DHCP server assigns a new IP address to the module, the PC must have some way of obtaining the module's new IP address. The **IP Address report** fields are used to set up periodic reporting of the module's IP address when the module's IP address is assigned by a server. The IP address report is sent automatically at regular intervals to an IP address and TCP port number of your choice. There are two ways to view the module's IP address report on a PC:

- Install NPort Search Utility and use the built-in IP Address Report function.
- Develop software that parses the IP address report data. Please refer to Appendix C for details on how to parse the module's IP address report data.

Auto report to

Setting	Factory Default	Necessity
E.g., 192.168.1.1 or URL (IP addresses of the form x.x.x.0 and x.x.x.255 are invalid.)	None	Optional

Auto report to TCP port

Setting	Factory Default	Necessity
E.g., 4001	4002	Optional

If left blank, auto IP reporting is disabled. If an IP address is entered along with a TCP port number, the IP address reports will be sent to the specified address and port number.

Auto report period

Setting	Factory Default	Necessity
Time interval (in seconds)	10	Optional

The Auto report period field specifies how often the module will report its IP address. An auto report period of 10 seconds means that an IP address report will be sent every 10 seconds.

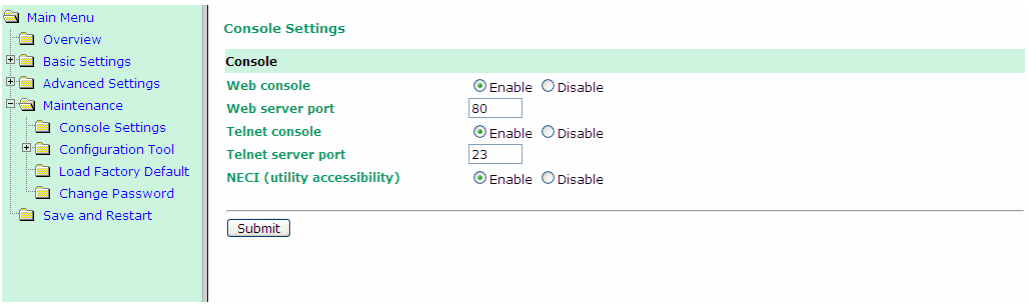
ATTENTION

The module will send IP address reports only when assigned an IP address from a DHCP or BOOTP server. If a connection to a DHCP or BOOTP server is not available, no IP address report will be sent.

Maintenance

Console Settings

You can enable or disable the **Web, Telnet, and Utility** console functions on this page. You can also set up the **Web server port** number and **Telnet server port** number here.

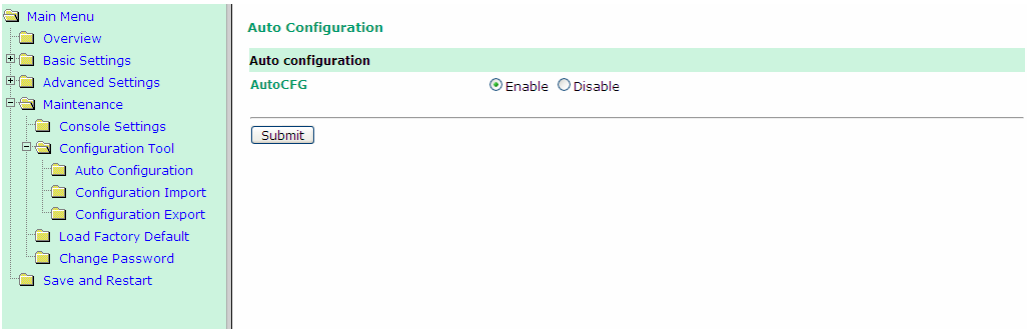


Configuration Tool

To enhancement your configuration efficiency, Moxa's MiiNePort E1 has built-in Auto Configuration, Configuration Import, and Configuration Export functions.

Auto Configuration

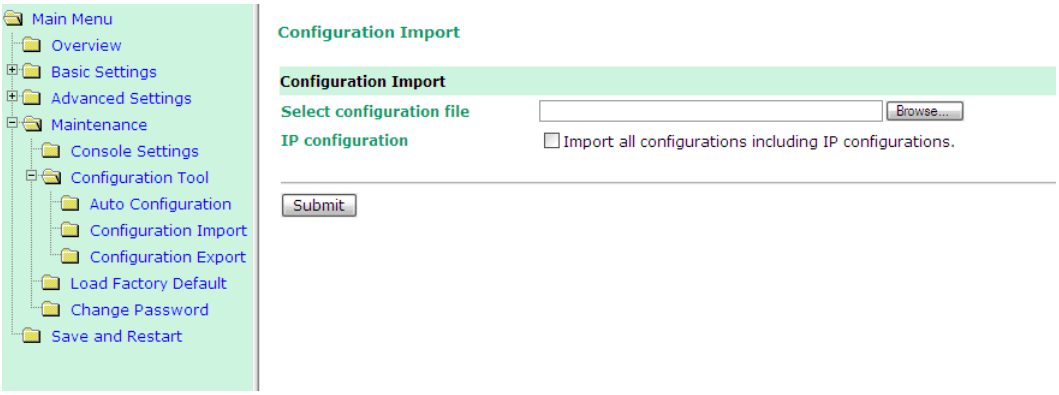
You can enable or disable the AutoCFG (Auto Configuration) function here. AutoCFG is designed for enabling automatic network configuration during your device production. By using the AutoCFG, you can realize real device mass production without a need to set up the network module one by one.



For more information about AutoCFG, please refer to Chapter 8, NetEZ Technologies' AutoCFG (Auto Confuguration).

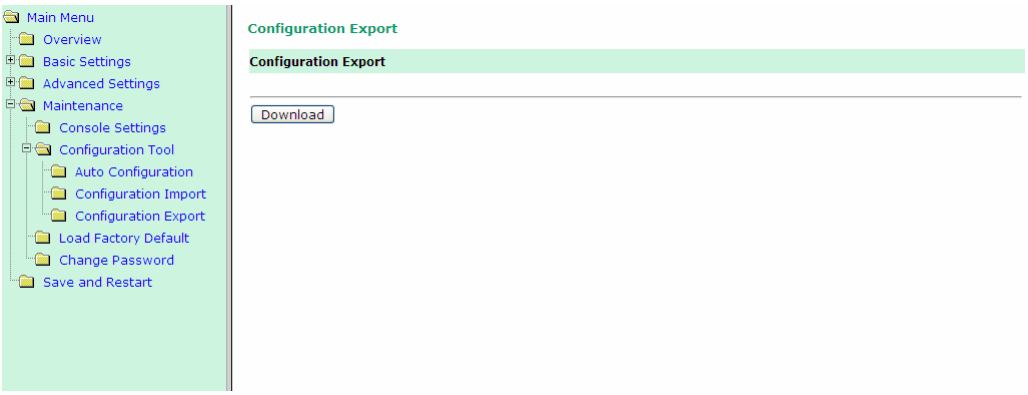
Configuration Import

You can select the proper cfg.txt file and then import it to instantly replicate the configuration to the current module. Also, you can choose whether to import IP configurations at the same time.



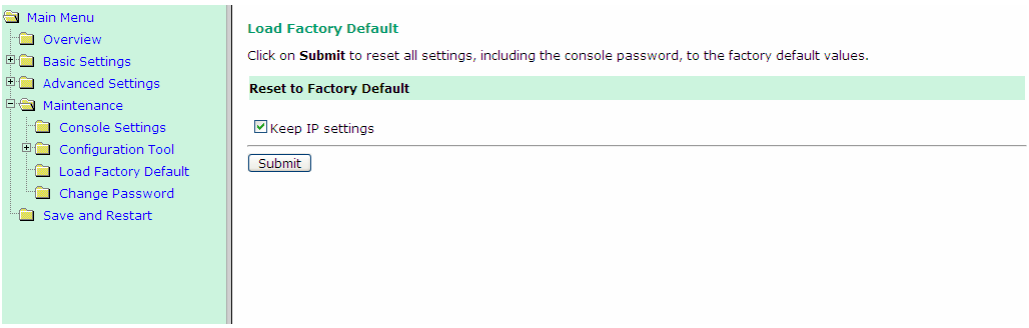
Configuration Export

You can download the current configuration as a **cfg.txt** file by clicking on Configuration Export. The configuration cfg.txt file can be later used in Configuration Import for effortless configuration replication or AutoCFG for mass configuration deployment during your device's mass production.



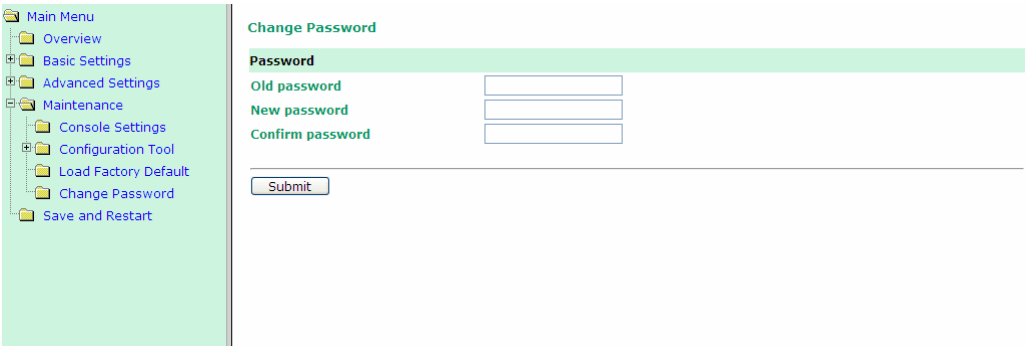
Load Factory Default

To load the factory default settings, click on **Load Factory Default** in the navigation panel and then click on **Submit**. All previous modifications will be lost, but you can choose to keep the IP settings by checking **Keep IP settings**.



Change Password

Click on **Change Password** in the navigation panel to display the **Change Password** window.



Enter the old password and new passwords and click on **Submit**. To remove password protection, leave the new password blank and just enter the old password.

ATTENTION

If you set a password for your MiiNePort E1 but forget it, you may want to reset the module to factory default. As you may not be able to operate the function through any software interface, we design 2 hardware tools for you to solve this problem;

1. Pin 7 on the Module: To use the "Load Factory Defaults" function, refer to Chapter 1, Pin Assignments. Check the Pin 7's definition. If you pull low Pin 7 for longer than 100ms, then pull high and the MiiNePort E1 will automatically load factory default settings.
2. EXTrigger: By pressing the EXTrigger when the device is powering on, the MiiNePort E1 module will reset itself to factory default settings. For details, please refer to Chapter 8, NetEZ Technologies.

8

NetEZ Technologies

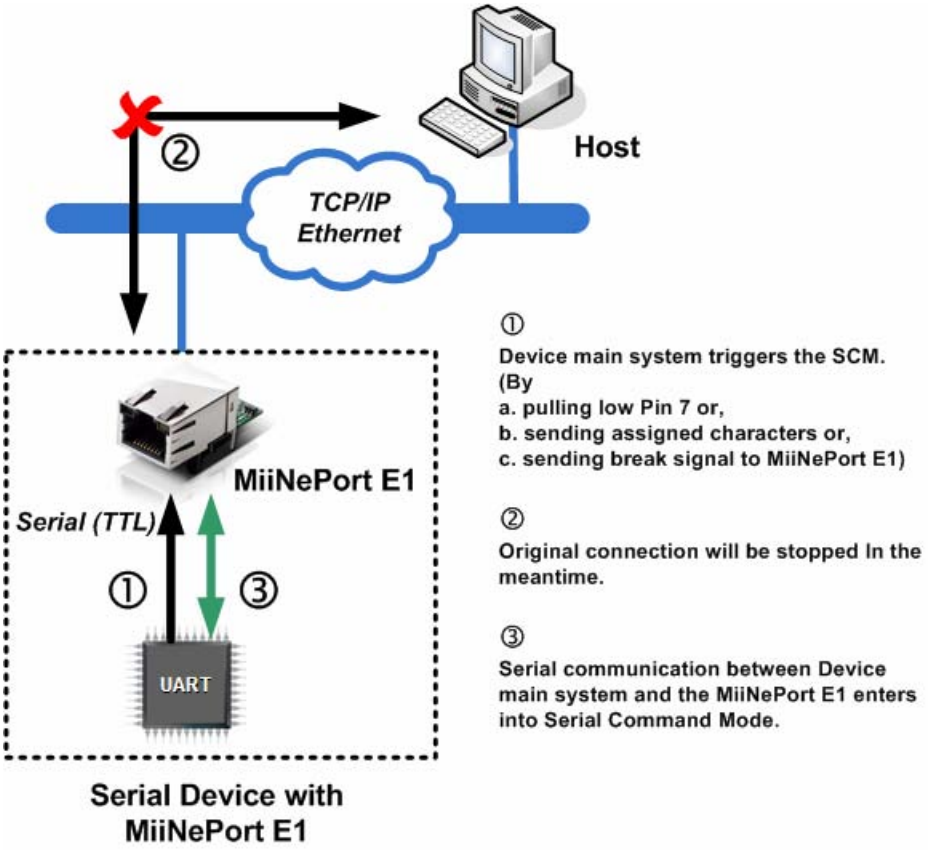
This chapter introduces the NetEZ technology family and its four innovative functions:

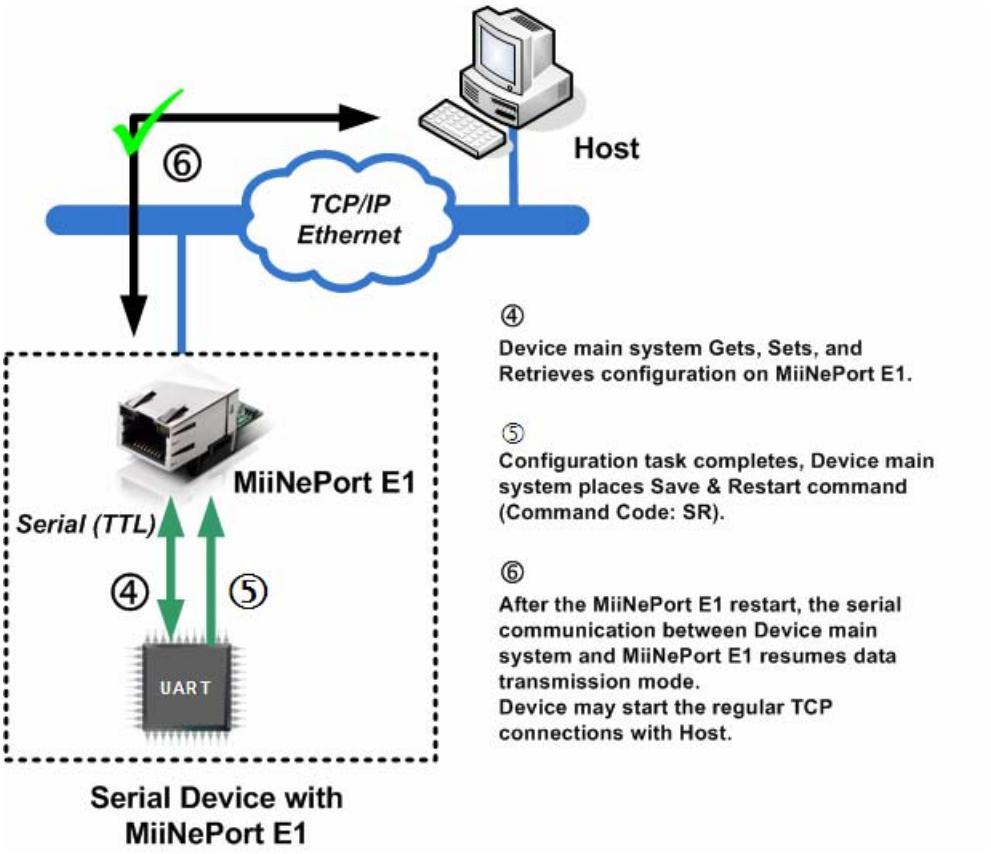
- **SCM (Serial Command Mode)**
- **EXTrigger (External Trigger)**
- **AutoCFG (Auto Configuration)**
- **MCSC (Multiple Channel Serial Communication)**

SCM (Serial Command Mode)

MiiNePort E1's SCM (Serial Command Mode) allows the module's parameters to be retrieved or configured through the serial port, rather than over the network. This is done through the use of specially parsed commands sent to the module through the serial port.

SCM is often used when your device has already been used in real application and the configuration change on the MiiNePort E1 is required, such as changing the device's IP address by the operation on your device's key pad. Please refer to the illustrated process flow below:




Host
⑥
TCP/IP
Ethernet
MiiNePort E1
Serial (TTL)
④
⑤
UART
Serial Device with
MiiNePort E1
④
Device main system Gets, Sets, and Retrieves configuration on MiiNePort E1.
⑤
Configuration task completes, Device main system places Save & Restart command (Command Code: SR).
⑥
After the MiiNePort E1 restart, the serial communication between Device main system and MiiNePort E1 resumes data transmission mode.
Device may start the regular TCP connections with Host.

Please refer to **Chapter 7: Web Console Configuration** for the most appropriate way to set SCM for your application.

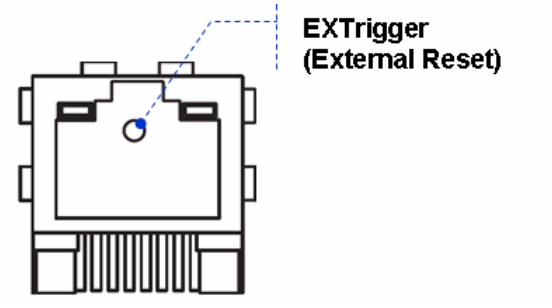
Please refer to Appendix A for detailed instructions for setting SCM commands.

EXTrigger (External Trigger)

The MiiNePort E1's **EXTrigger** has 2 functions; resetting the module to Moxa's default configuration and restarting the module. The **EXTrigger** is designed to provide you with an easy network troubleshooting tool without the need to stop the device's normal operation.

The **Reset to Default** function is enabled by default and cannot be disabled. This design prevents device deadlock in the future if you or your customer forgets the password and is unable to access the console to troubleshoot the device. To activate the **EXTrigger's Reset to Default** function, you just need to press the button inside the RJ45 jack for 5 seconds while your device, together with the MiiNePort E1, is starting up.

The **Restart** function [18] [19] simply shuts down the module and then turns it back on. This does not affect any settings in the module but is a commonly used method for initial troubleshooting. Simply hold the button for 5 seconds at any time during device operation to restart the MiiNePort E1 module.



Please refer to the following table describing the LED interaction with the EXTrigger.

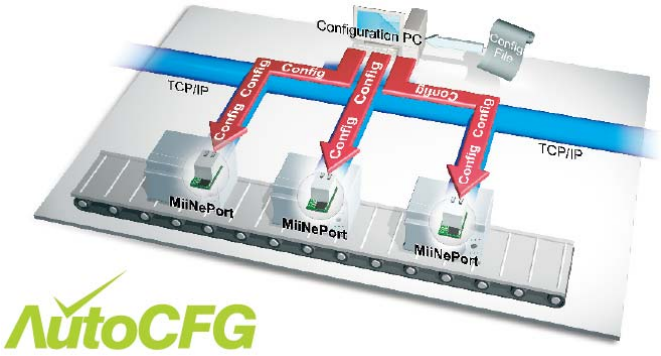
EXTrigger	Fault/In Use LED Behavior	Blinking Duration	Trigger Timing	Note
Reset to Default	1 time/sec (Blinks amber quickly)	5 sec	Press the button, then power on the device.	When LED is Off (stops blinking) 1. Function active 2. Release button
Restart	0.5 time/sec (Blinks amber slowly)	5 sec	Press the button anytime during operation.	

Furthermore, the EXTrigger is actually an external Digital Input (DI) which can be accessed by you or your device's end user. You can design your own application by leveraging this external DI. Please contact Moxa for any customized functions on EXTrigger.

[18] Please refer to Chapter 7, Web Console Configuration, Pin and IO Settings, External Reset Function to enable EXTrigger of your MiiNePort E1.

[19] The **Restart** function of ExTrigger is disabled in Moxa's factory default settings.

AutoCFG (Auto Configuration)



MiiNePort E1's AutoCFG [20] is designed for realizing the true mass production of your device. With AutoCFG, you no longer need to set up the network module one by one during the device manufacturing process. To take advantage of AutoCFG, please follow the steps described below:

Step 1: Complete the appropriate configuration on one of your MiiNePort E1 modules.

Step 2: Export the configuration. Please refer to the Chapter 7: Web Console Configuration, Maintenance, Configuration Tool, Configuration Export. After you export your configuration, save a configuration file named MiiNePort E1.cfg, which contains all your settings for the MiiNePort E1 module.

Step 3: Establish a TFTP server on the network where you plan to conduct the AutoCFG task on your device production line. Set your TFTP server's IP address as 169.254.x.x/16. Then, save the previously saved **MiiNePort E1.cfg** file under the root directory C:\ of the TFTP server [21]. Now the AutoCFG working environment is ready.

Step 4: Plug in the RJ45 cable into the MiiNePort E1 modules.

Step 5: Your MiiNePort E1 modules would then be automatically assigned temporary IP addresses by AutoIP protocol.

Step 6: Your MiiNePort E1 module will actively send out a broadcast packet asking if there is a TFTP server that has a file named **MiiNePort E1.cfg** in the same subnet.

Step 7: Your TFTP server will respond to the broadcast packet and your MiiNePort E1 modules will automatically download the **MiiNePort E1.cfg** from the TFTFP server.

Step 8: Your MiiNePort E1 modules will then import the configuration, finish the configuration, save the configuration and then restart. Once the MiiNePort E1 finishes the self-configuration, the Fault/In-Use LED on the RJ45 connector will blink with Green and Amber in an interlacing manner. When you see the LED blinking, AutoCFG task is completed and you can restart your device or Power Off it.

[20] AutoCFG is set enabled in Moxa's factory default settings. If you want to disable AutoCFG, please refer to Chapter 7, Web Console Configuration, Maintenance, Configuration Tool, Auto Configuration.

[21] You can find many freeware on the internet that can help you easily establish your TFTP server.

Step 9: When you need to change the configuration, just complete the revised configuration, export the new **MiiNePort E1.cfg**, save the file onto the established TFTP server, choose to replace the original **MiiNePort E1.cfg** under the root directory C:\. Your MiiNePort E1 modules will then be able to retrieve the correctly revised configuration.

ATTENTION

If AutoCFG is not working properly on your device, check if the:

1. AutoCFG function is enabled (the function is enabled by default)
2. TFTP Server is working properly
3. Specific configuration file name is **MiiNePort E1.cfg**

ATTENTION

Theoretically there is no quantity limitation of the MiiNePort E1 modules to be placed on one subnet to conduct the AutoCFG task at same time.

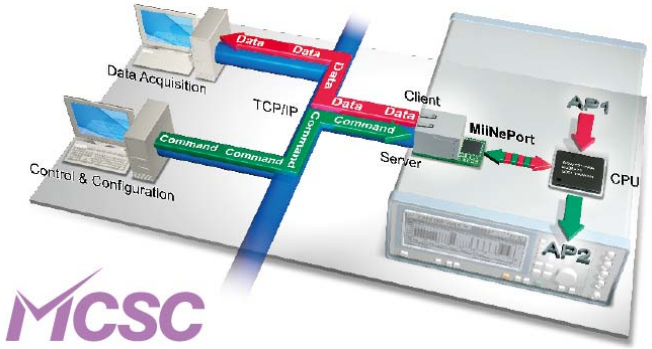
ATTENTION

Even if you don't want the auto configuration function, the module will still go through the process of searching the cfg.txt file on the network every time it starts up if you don't disable the AutoCFG function on your MiiNePort E1. The MiiNePort E1 will resume normal operation if it cannot find the correct TFTP server with the **cfg.txt** file on the network. The time length for the search will not exceed 5 seconds on average, and your MiiNePort E1 will work properly with its original configuration even if it doesn't complete the AutoCFG task.

ATTENTION

If you don't want the AutoCFG function to stay alive after you've successfully finished the auto-configuration, we suggest you DISABLE the AutoCFG function when you are making your MiiNePort E1.cfg. That way you won't see the AutoCFG activity recur the next time your device is powered on. Please refer to **Chapter 7: Web Console Configuration** for configuration details.

MCSC (Multiple Channel Serial Communication)



Our Motivation for Providing MCSC

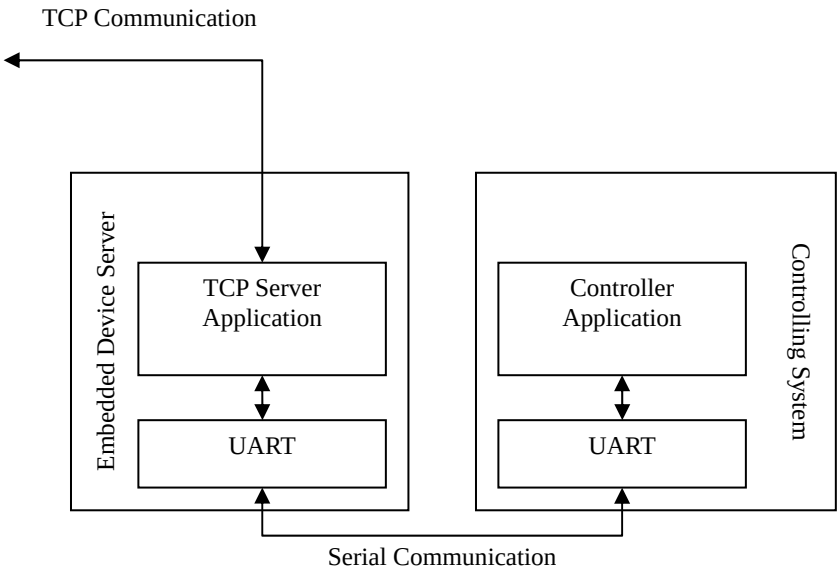
An embedded device server only provides a limited number of physical serial ports to communicate with the device's main system. As a result, users with one physical serial port [22] are generally limited to a single application. If you want to implement a second, third, or more serial-to-Ethernet applications, additional physical serial ports are needed. Unfortunately, providing additional physical serial ports equals more hardware costs. Moxa's MiiNePort E1's MCSC is designed to solve this dilemma. MCSC (Multiple Channel Serial Communication) is a lightweight, pure software solution that enables multiple serial-to-Ethernet applications all through one physical serial port without any incremental hardware costs to you. In short, MCSC allows multiple application channels to share a single physical serial port at the same time.

Communication Model

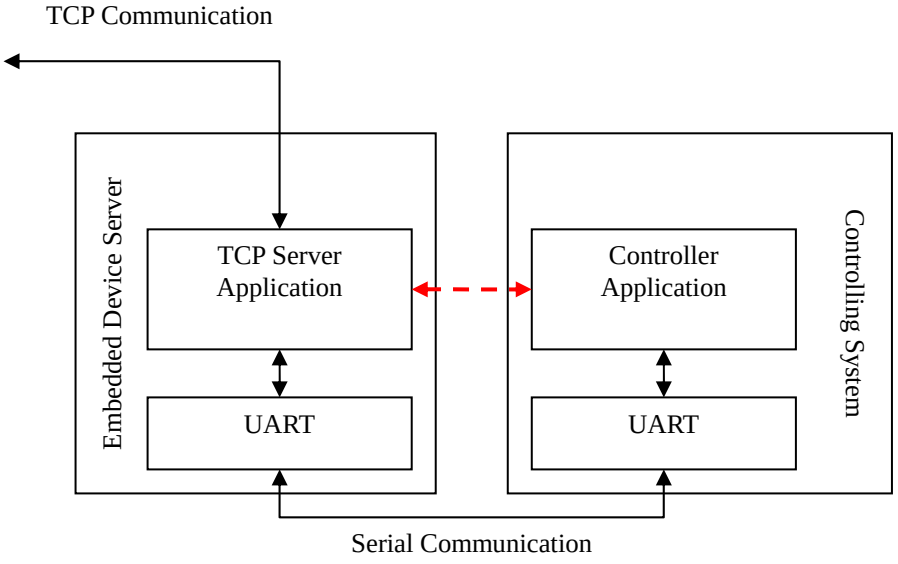
In traditional serial communication models, communication is accomplished by connecting two devices over a single serial channel. At the embedded device server end, Operating Mode (OP Mode) enables the other end of the serial channel to transmit or receive data from the network. However, only one application, TCP Server transportation, TCP Client transportation, or UDP transportation [23] is supported at a time. The communication model is depicted in the following figure:

[22] In the case of the MiiNePort E1, there is only one physical serial port provided.

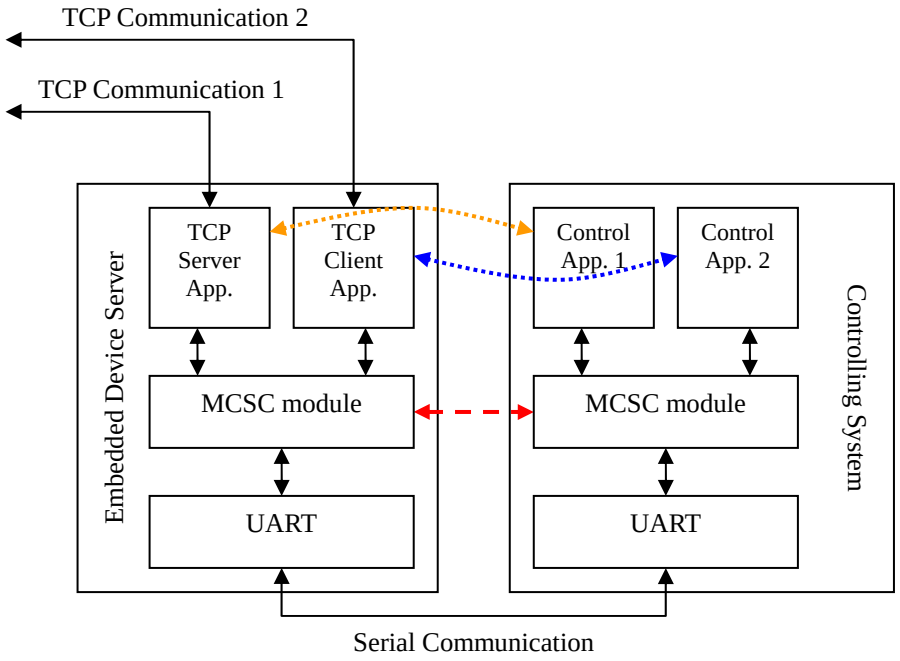
[23] Please refer to Chapter 3, Choosing the Proper Operation Mode for the introduction on the Operation Modes provided by MiiNePort E1.



Moreover, although there are UARTs, and even OS and driver stacks, between the TCP Server Application and Controller Application, the applications are considered to be connected by a logical bi-directional channel indicated in red in below figure;



When using MCSC, each serial communication end is attached with an MCSC command encoding and decoding module[24]. This module is used primarily for UART applications. It compresses the data transmitted by the application to the single physical serial channel, and dispatches the data received from the physical serial channel to the appropriate onward applications. With the help of the MCSC modules, the two applications at opposite ends are still connected to a logical bi-directional channel, but two or more application pairs are also connected by separate channels to the physical serial channel. The communication model is depicted in the following figure.



The MCSC module has been built into the MiiNePort E1 series products. To enable MCSC, you need to set your MiiNePort E1 serial port operation mode to MCSC mode and then set the channels' operation mode individually according to your application. In addition, you also need to implement the MCSC module in your device's main system.

There are some principles for the MCSC module implementation.

1. MCSC uses a command-based stream protocol. That is, all transmissions that do not fit MCSC command format are treated as application data. All MCSC commands begin with ASCII DLE characters. For the detailed command format, please refer to the Command Packets section in following section.
2. Other than dealing with MCSC commands, an MCSC module simply transfers application data from the upper application channel to the underlying physical serial channel, and vice versa.
3. Like traditional serial applications, MCSC modules are logically connected to a bi-directional channel. To use the serial bandwidth more effectively, MCSC modules separate it into two uni-directional channels. Each MCSC module maintains its transmission application channel and reception application channel. More clearly, the transmission channel of one end is the reception channel of another.

[24] Module here means software programming module concept.

4. Although logically all upper application channels share one physical serial port, only one channel can be connected to the physical serial port to prevent confusion. More clearly, the application channel connected to the transmission channel is called the active transmission channel, which transmits data to the serial channel. The application channel connected to the reception channel is called the active reception channel, which receives data from the serial channel.
5. An MCSC module decides the active transmission channel. In other words, the active reception channel is decided by the other MCSC module lying at the other end of the serial communication channel.
6. To avoid data loss, we recommend that MCSC is applied with flow control over serial communication.

Command Packets

An MCSC control packet begins with an ASCII DLE (Data Link Escape, 0x10) character. An end of serial communication (normally, an MCSC module) treats the DLE it receives as a special delimiter that indicates the other end of serial communication (normally, another MCSC module) that attempts to query or change the MCSC behavior.

So if a serial communication end wants to send a 0x10 (DLE) data byte, it sends two DLEs, one after another. The receiving end decodes these two DLEs into a single 0x10 byte.

Currently, there are 4 kinds of control packets defined under MCSC:

Channel Switch Command

DLE 0x10	**SOH** 0x01	***CHN*** 0x??

This command is used for an end to switch its active transmission channel. The **CHN** field indicates the new channel number. Note that the **CHN** index is zero based, so you'll need to use 0x00 for **CHN** to switch to channel 1, 0x01 to switch to channel 2, and so on. For safety reasons, if one serial communication end receives a Channel Switch Command that switches to a channel it does not support, it will simply discard it after data transmission.

Note that at the beginning of MCSC, both active communication channels are assumed to be channel 1. An end of serial communication transfers all data without directing Channel Switch Command to channel 1 application.

In the MiiNePort E1, SCM will always be enabled when MCSC is enabled. A special channel indexed as 0xFF in MCSC is dedicated for SCM. For more information, refer to the Serial Command Mode section in this chapter.

Example:

A>	10	01	01	11	22	33										10	01	00	44
B>									10	01	01	33	22	11					
A>	55	66																	
B>			11	22	33	10	01	00	66	55	44	44	55	66					

This example shows that end A sends 3 bytes of data (11, 22, 33) using channel 1 and 3 bytes of data (44, 55, 66) using channel 2. At the other side, end B sends 6 bytes of data (33, 22, 11, 11, 22, 33) using channel 1 and 6 bytes of data (66, 55, 44, 44, 55, 66) using channel 2.

Channel Enquiry Command

DLE 0x10	ENQ 0x05

This command is used for an end to query for its active reception channel. By receiving this command, the other end shall resend a Channel Switch Command to indicate the active transmission channel it is dealing with, that is, the active reception channel which the initiator of Channel Enquiry Command deals with.

Example:

A>	10	05															
B>			10	01	01	11	22	33									

This examples shows that end A queries for the active transmission channel of end B, end B replies for channel 2 and then continues to send 3 bytes of data (11, 22, 33) using channel 2.

Data Escape Command

DLE 0x10	DLE 0x10

This is not actually an MCSC command. Since the DLE character is used by MCSC to escape its commands, a pair of DLEs shall be transmitted to indicate a single 0x10 character of data.

Example:

A>	10	01	00	00	02	04	06	08	0A	0C	0E	10	10	12			
B>																	

This examples shows that the end A sends 10 bytes of data (00, 02, 04, 06, 08, 0A, 0C, 0E, 10, 12) using channel 1.

Abnormal Packets

DLE 0x10	OTHER ???

Once the MiiNePort E1 receives a data stream with a header DLE character followed by characters that are not SOH, ENG, or DLE characters, the MiiNePort E1 will see this as an abnormal data packet caused by communication problems and will drop this data packet without sending out through Ethernet. In the meantime MiiNePort E1 will return the channel enquiry command (10 05) to the serial main system through serial communication helping the main system to detect and trouble shoot the problem.

SCM (Serial Command Mode) under MCSC

When MCSC is enabled, a special application channel indexed as 0xFF (channel 256) is dedicated for Moxa SCM without additional configuration required. SCM enables the controlling system to configure, monitor, or control the attached embedded device server. For more information about what SCM can do, please refer to the Serial Command Mode chapter.

With MCSC, normal network communication continues uninterrupted when the controlling system is monitoring or diagnosing the embedded device server with SCM, providing more flexibility for SCM.

A

SCM (Serial Command Mode) Command Set Introduction

Command/Reply Format

Single Line Command Format

Head 1 byte	Op 1 byte	Cmd 2 bytes	Parameters 0 to n bytes	Tail 1 or 2 bytes

Single Line Reply Format

Head 1 byte	Op 1 byte	Cmd 2 bytes	ST 1 byte	Parameters 0 to n bytes	Tail 1 or 2 bytes

Head, and Tail

	Head 1 byte		**Tail** 1 or 2 bytes
Command	?		CR LF CR-LF
Reply	!		LF

Operation Codes

Operation Code	Meaning
G	Get configuration from MiiNePort's RAM
S	Set configuration to MiiNePort's RAM
R	Retrieve running configuration
V	View status
C	Control

Status Codes

Status Code	Meaning
E	Enter Serial Command Mode
0	Command was executed successfully
1	Unrecognized format
2	Operation is not valid
3	Command is not valid
4	Parameter is incorrect
5	Parameter is too long

Restriction

The total amount of parameters within a single command must not exceed 1024 characters.

Command Codes for Getting Configuration

Device Name

Command code: **BN**

Command parameters: N/A

Reply parameters: MiiNePort's name.

?GBN↵ !GBN0MiiNePort_9527↵	System requests configured device name for this MiiNePort. MiiNePort reports device name as 'MiiNePort_9527'.

Console Password

Command code: **BP**

Command parameters: N/A

Reply parameters: MiiNePort's console password.

?GBP↵ !GBP01234↵	System requests console password for this MiiNePort. MiiNePort reports console password as '1234'.

Web console

Command code: **BH**

Command parameters: N/A

Reply parameters: 1 and web console TCP port separated by a semicolon (;) if web console is enabled, or a 0 if it is disabled.

?GBH↵ !GBH01;80↵	System requests http console setting for this MiiNePort. MiiNePort reports http console as 'Enable' and http port as '80'.

Telnet console

Command code: **BT**

Command parameters: N/A

Reply parameters: 1 and telnet console TCP port separated by a semicolon (;) if telnet console is enabled, or a 0 if it is disabled.

?GBT↵ !GBT01;23↵	System requests telnet console setting for this MiiNePort. MiiNePort reports telnet console as 'Enable' and telnet port as '23'.

Auto configuration

Command code: **BA**

Command parameters: N/A

Reply parameters: 1 if Auto configuration is enabled, otherwise reply 0.

?GBA↵ !GBA01↵	System requests Auto configuration setting for this MiiNePort. MiiNePort reports Auto configuration as 'Enable'.

External reset function

Command code: **BE**

Command parameters: N/A

Reply parameters: 1 if external reset function is enabled, otherwise 0

?GBE↵ !GBE01↵	System requests external reset function setting for this MiiNePort. MiiNePort reports Auto configuration as 'Enable'.

NECI (utility accessibility)

Command code: **BU**

Command parameters: N/A

Reply parameters: 1 if NECI is enabled, otherwise 0.

?GBU↵ !GBU01↵	System requests NECI setting for this MiiNePort. MiiNePort reports NECI as 'Enable'.

IP configuration

Command code: **NC**

Command parameters: N/A

Reply parameters: MiiNePort's IP configuration index as following.

0	Static
1	DHCP
2	DHCP/BOOTP
3	BOOTP
4	AUTOIP

?GNC↵ !GNC00↵	System requests IP configuration for this MiiNePort. MiiNePort reports IP configuration as 'Static'.

IP address

Command code: **NI**

Command parameters: N/A

Reply parameters: MiiNePort's IP address.

?GNI↵ !GNI0192.168.127.254↵	System requests IP address for this MiiNePort. MiiNePort reports IP address as '192.168.127.254'.

Netmask

Command code: **NM**

Command parameters: N/A

Reply parameters: MiiNePort's Netmask address.

?GNM↵ !GNM0255.255.255.0↵	System requests Netmask address for this MiiNePort. MiiNePort reports netmask as '255.255.255.0'.

Gateway

Command code: **NG**

Command parameters: N/A

Reply parameters: MiiNePort's Gateway address.

?GNG↵ !GNG0255.255.255.255↵	System requests Gateway address for this MiiNePort. MiiNePort reports Gateway address as '255.255.255.255'.

DNS

Command code: **ND**

Command parameters: The index (1 or 2) of DNS server.

Reply parameters: MiiNePort's DNS address.

?GND1↵ !GND0192.168.1.2↵	System requests DNS server 1 address for this MiiNePort. MiiNePort reports DNS server 1 address as '192.168.1.2'.

PHY Speed

Command code: **NS**

Command parameters: N/A

Reply parameters: MiiNePort's PHY speed as following:

0	PHY auto negotiation is enabled.
10	PHY speed is forced to 10Mbps with half duplex
11	PHY speed is forced to 10Mbps with full duplex
100	PHY speed is forced to 100Mbps with half duplex
101	PHY speed is forced to 100Mbps with full duplex

?GNS↵ !GNS00↵	System requests PHY speed setting for this MiiNePort. MiiNePort reports PHY speed determined by PHY auto negotiation function.

Enable/Disable SNMP

Command code: **MS**

Command parameters: N/A

Reply parameters: 1 (Enable) or 0 (Disable) MiiNePort's SNMP agent.

?GMS↵ !GMS01↵	System requests Enable/Disable SNMP agent for this MiiNePort. MiiNePort reports SNMP agent as 'Enable'.

Community String

Command code: **MU**

Command parameters: N/A

Reply parameters: MiiNePort's SNMP Community string.

?GMU↵ !GMU0public↵	System requests SNMP community string for this MiiNePort. MiiNePort reports SNMP community string as 'public'.

Contact Name

Command code: **MN**

Command parameters: N/A

Reply parameters: MiiNePort's SNMP Contact Name.

?GMN↵ !GMN0s_name↵	System requests SNMP contact name for this MiiNePort. MiiNePort reports SNMP contact name as 's_name'.

Location

Command code: **ML**

Command parameters: N/A

Reply parameters: MiiNePort's SNMP Location.

?GML↵ !GML0s_location↵	System requests SNMP location for this MiiNePort. MiiNePort reports SNMP location as 's_location'.

Enable/Disable Accessible IP List

Command code: **AS**

Command parameters: N/A

Reply parameters: Enable (1) or Disable (0) MiiNePort's accessible IP list.

?GNS↵ !GNS01↵	System requests Enable/Disable accessible IP list for this MiiNePort. MiiNePort reports accessible IP list as 'Enable'.

Accessible IP List

Command code: **AI**

Command parameters: The index of accessible IP list (from 1 to 16)

Reply parameters: Return format Mode;IP;Netmask in the accessible IP list. If mode equals to 1, it's active, otherwise it's inactive.

?GAI1↵ !GAI01;192.168.1.2;255.255.255.0 ↵	System requests 1st accessible IP list for this MiiNePort. MiiNePort reports 1st accessible IP list as 'Active', IP address as '192.167.1.2', and Netmask as '255.255.255.0'.

Auto IP Report

Command code: **NR**

Command parameters: N/A

Reply parameters: MiiNePort's auto IP report setting.

?GNR↵ !GNR0192.168.1.250:4000;50↵	System requests auto IP report for this MiiNePort. MiiNePort reports Auto IP report server as '192.168.1.250' and port as '4000', and report period as '50'.

Port alias

Command code: **SA**

Command parameters: Port index.

Reply parameters: MiiNePort's Port alias.

?GSA1↵ !GSA0port1↵	System requests port 1 alias for this MiiNePort. MiiNePort reports Port alias as 'port1'.

Baud rate

Command code: **SB**

Command parameters: Port index.

Reply parameters: MiiNePort's Baud rate.

?GSB1↵ !GSB0115200↵	System requests port 1's Baud rate for this MiiNePort. MiiNePort reports Baud rate as '115200'.

Data bits

Command code: **SD**

Command parameters: Port index.

Reply parameters: MiiNePort's Data bits.

?GSD1↵ !GSD08↵	System requests port 1's Data bits for this MiiNePort. MiiNePort reports Data bits as '8'.

Stop bits

Command code: **SS**

Command parameters: Port index.

Reply parameters: MiiNePort's stop bits index.

Stop Bits Index Table

0	Stop bits is 1
1	Stop bits is 1.5
2	Stop bits is 2

?GSS1↵ !GSS00↵	System requests port 1's Stop bits for this MiiNePort. MiiNePort reports Stop bits as '1'.

Parity

Command code: **SP**

Command parameters: Port index.

Reply parameters: MiiNePort's Parity index as follows.

0	None
1	Odd
2	Even
3	Mark
4	Space

?GSP1↵ !GSP00↵	System requests port 1's Parity for this MiiNePort. MiiNePort reports Parity as 'None'.

Flow control

Command code: **SL**

Command parameters: Port index.

Reply parameters: MiiNePort's Flow control index as follows.

0	None
1	RTS/CTS
2	XON/XOFF
3	DTR/DSR

?GSL1↵ !GSL01↵	System requests port 1's Flow control for this MiiNePort. MiiNePort reports Flow control as 'RTS/CTS'.

FIFO

Command code: **SF**

Command parameters: Port index.

Reply parameters: 1 (Enable) or 0 (Disable) MiiNePort's FIFO.

?GSF1↵ !GSF01↵	System requests port 1's FIFO for this MiiNePort. MiiNePort reports FIFO as 'Enable'.

Operation Mode

Command code: **OM**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For port itself, channel index shall be 0.

Reply parameters: MiiNePort's operation mode index as shown in the following table.

0	Disable
1	RealCom
2	TCP
3	UDP
4	MCSC (This value is unavailable for MCSC channels.)

?GOM1;0↵ !GOM04↵ ?GOM1;2↵ !GOM02↵	System requests port 1's Operation Mode for this MiiNePort. MiiNePort reports Operation Mode as 'MCSC'. When port 1 is set to MCSC, system requests the operation mode for channel 2 of port 1. MiiNePort reports operation mode as `TCP'.

Data Packing Length

Command code: **OL**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's data packing length as follows.

?GOL1;0↵	System requests port 1's data packing length for this MiiNePort.
!GOL0256↵	MiiNePort reports data packing length as '256'.
?GOL1;2	When port 1 is set to MCSC, system requests the data packing length for channel 2 of port 1.
!GOL0128	MiiNePort reports data packing length as `128'.

Delimiter

Command code: **OD**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's delimiter setting. (delimiter1 enable/disable; hex1; delimiter2 enable/disable; hex2)

?GOD1;0↵	System requests port 1's delimiter setting for this MiiNePort.
!GOD1;10;1;13↵	MiiNePort reports delimiter 1 as 'Enable' and hex code as '10', delimiter 2 as 'Enable' and hex code as '13'.

Match Bytes

Command code: **OY**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's match bytes.

?GOY1;0↵	System requests port 1's match bytes for this MiiNePort.
!GOY02↵	MiiNePort reports match bytes as '2'.

Delimiter Process

Command code: **OT**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's delimiter process.

0	Do nothing
1	Delimiter+1
2	Delimiter+2

?GOT1;0↵ !GOT02↵	System requests port 1's delimiter process for this MiiNePort. MiiNePort reports delimiter process as 'Delimiter+2'.

Force Transmit

Command code: **OF**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's force transmit timeout.

?GOF1;0↵ !GOF00↵	System requests port 1's force transmit timeout for this MiiNePort. MiiNePort reports force transmit timeout as '0' sec.

RealCOM TCP alive check time

Command code: **RA**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's TCP alive check time.

?GRA1;0↵ !GRA05↵	System requests port 1's TCP alive check time for port 1. MiiNePort reports TCP alive check time as '5' min.

RealCOM max connection

Command code: **RM**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's TCP maximum connection number.

?GRM1;1↵ !GRM04↵	System requests maximum connection number for port 1 and channel 1. MiiNePort reports maximum connection number as '4'.

RealCOM Ignores Jammed IP

Command code: **RJ**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GRJ1;2↵ !GRJ01↵	System requests jammed IP policy for port 1's channel 2. MiiNePort reports ignore jammed IP as 'Enable'.

RealCOM Allows Driver Control

Command code: **RD**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GRD1;0↵ !GRD01↵	System requests allow driver control policy for port1. MiiNePort reports allow driver control as 'Enable'.

TCP Role

Command code: **TO**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP role as shown in the following table.

0	TCP server
1	TCP client
2	Mixed

?GTO1;0↵ !GTO00↵	System requests TCP role for port 1. MiiNePort reports TCP role as TCP server.

TCP Server Connection Control

Command code: **TS**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP server connection control as shown in the following table.

0	Always accept
1	Accept with DSR on

?GTS1;0↵ !GTS00↵	System requests TCP server connection control for port 1. MiiNePort reports incoming policy as 'Always accept'.

TCP Password

Command code: **TW**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Require) or 0 (Don't require) and password.

?GTW1;0↵ !GTW01;1234↵	System requests port 1's password setting for port 1. MiiNePort reports password as 'required' and password as '1234'.

TCP Communication Protocol

Command code: **TR**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 0 (Raw TCP) or 1 (RFC 2217)

?GTR1;0↲ !GTR00↲	System requests communication protocol for port 1. MiiNePort reports communication protocol as 'Raw TCP'.

TCP Alive Check Time

Command code: **TA**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP alive check time

?GTA1;0↲ !GTA05↲	System requests TCP alive check time for port 1. MiiNePort reports TCP alive check time as '5' minutes.

TCP Port

Command code: **TP**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP port

?GTP1;0↲ !GTP04100↲	System requests TCP port for port 1. MiiNePort reports TCP port as '4100'.

TCP Client Connection Control

Command code: **TC**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP client connection control index as shown in the following table

0	Any character
1	Manual connection
2	Start up
3	With DSR on

?GTC1;0↵ !GTC01↵	System requests TCP client connection control for port 1. MiiNePort for TCP client connection control as 'Any character'.

TCP Connection Response

Command code: **TN**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GTN1;0↵ !GTN00↵	System requests connection response policy for port 1. MiiNePort reports connection response as 'Disable'.

TCP destination address

Command code: **TI**

Command parameters: Three numbers separated by a semicolon (;) denotes port index, MCSC channel index, and destination address index. For MCSC-disabled port, channel index shall be 0. The destination address index is as following:

0	Destination address
1	Alternated address 1
2	Alternated address 2
3	Alternated address 3

Reply parameters: TCP destination address and port

?GTI1;0;0↵ !GTI0192.168.1.2:4001↵	System requests destination address for port 1. MiiNePort reports destination address as '192.168.1.2' and port as '4001'.

TCP connect retry timeout

Command code: **TT**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP connect retry timeout

?GTT1;0↵ !GTI01500↵	System requests TCP connect retry timeout for port 1. MiiNePort reports TCP connect retry timeout as '1500'.

TCP max connection

Command code: **TM**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's TCP maximum connection number.

?GTM1;1↵ !GTM04↵	System requests maximum connection number for port 1 and channel 1. MiiNePort reports maximum connection number as '4'.

TCP ignore jammed IP

Command code: **TJ**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GTJ1;2↵ !GTJ01↵	System requests jammed IP policy for port 1's channel 2. MiiNePort reports ignore jammed IP as 'Enable'.

TCP disconnect with DSR off

Command code: **TL**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GTL1;2↵ !GTL01↵	System requests DSR off policy for port 1's channel 2. MiiNePort reports DSR off as 'Enable'.

TCP check EOT character

Command code: **TE**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable), and EOT character.

?GTE1;0↵ !GTE01;04↵	System requests check EOT character setting for port 1. MiiNePort reports check EOT character as 'Enable' and EOT character as '0x04'.

TCP inactivity time

Command code: **TV**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP inactivity time.

?GTV1;0↵ !GTV00↵	System requests inactivity time for port 1. MiiNePort reports inactivity time as '0'.

UDP serial format

Command code: **UT**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: Serial format as following

0	Raw data
1	Packet command

?GUT1;0↲ !GUT00↲	System requests UDP serial format for port 1. MiiNePort reports UDP serial format as 'Raw data'.

UDP destination address

Command code: **UD**

Command parameters: Three numbers separated by a semicolon (;) denotes port index, MCSC channel index, and destination address index from 1 to 4. For MCSC-disabled port, channel index shall be 0.

Reply parameters: Begin address – End address: port

?GUD1;0;3↲ !GUD0192.168.1.3-1 92.168.1.8:4001↲	System requests UDP destination address 3 for port 1. MiiNePort reports UDP destination address is from '192.168.1.3' to '192.168.1.8' and port as '4001'.

UDP local listen port

Command code: **UP**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: local listen port

?GUP1;0↲ !GUP4001↲	System requests UDP local listen port for port 1. MiiNePort reports UDP local listen port as '4001'.

Pin function

Command code: **PF**

Command parameters: PIN index from 1 to 8.

Reply parameters: The PIN function is as following:

0	GND
1	VCC
2	RST
3	Tx
4	Rx
5	Ready/RTS
6	CTS
7	DTR
8	DSR
9	485_en
10	DIO
11	Reset to default

?GPF1↵	System requests PIN 1's function for this MiiNePort.
!GPF00↵	MiiNePort reports PIN function as 'GND'.
?GPF7↵	System requests PIN 7's function for MiiNePort.
!GPF011↵	MiiNePort reports PIN function is `Reset to default'.

DIO initial mode

Command code: **PM**

Command parameters: DIO port index. For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Reply parameters: 1 (output) or 0 (input), note that for MiiNePort E1, this configuration is available if specific PIN function is set to DIO.

?GPM1↵	System requests DIO1's initial mode for this MiiNePort.
!GPM01↵	MiiNePort reports DIO1's initial mode as 'output'.

DIO initial state

Command code: **PS**

Command parameters: DIO port index. For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Reply parameters: 1 (high) or 0 (low), note that this function is configuration only if specific DIO port is set to initial output.

?GPS0↵ !GPS00↵	System requests DIO0's initial state for this MiiNePort. MiiNePort reports DIO0's initial state as 'low'.

Serial command mode trigger

Command code: **CT**

Command parameters: N/A

Reply parameters: serial command mode trigger index as shown in the following table.

0	Disabe
1	H/W control pin(DIO1)
2	Activated by characters
3	Activated by break signal

Note that this configuration is valid only if port 1's operation mode is not set to MCSC.

?GCT↵ !GCT01↵	System requests serial command mode trigger method for this MiiNePort. MiiNePort reports serial command mode is triggered by DIO1.

Serial command mode trigger characters

Command code: **CC**

Command parameters: N/A

Reply parameters: serial command mode trigger characters, in hex, separated by semicolon (;).

Note that this configuration is valid only if serial command mode trigger method is set to `Activated by characters'.

?GCC↵ !GCC058;58;58↵	System requests serial command mode trigger characters for this MiiNePort. MiiNePort reports serial command mode triggered by `XXX' (0x585858).

Serial command mode trigger only at boot up

Command code: **CB**

Command parameters: N/A

Reply parameters: 0 if serial command mode can be triggered at any time, or 1 if it can only be triggered at boot up.

Note that this configuration is valid only if serial command mode trigger method is set to `Activated by characters'.

?GCB↵ !GCB01↵	System requests serial command mode for this MiiNePort. MiiNePort reports serial command mode can only be triggered by characters at boot up.

Gratuitous ARP

Command code: **VA**

Command parameters: N/A.

Reply parameters: 1 and send period if gratuitous ARP is enable; otherwise, reply 0.

?GVA↵ !GVA1;300↵	System requests gratuitous ARP status for alert mail server. MiiNePort reports gratuitous ARP as 'Enable' and send period as '300' sec.

Command Codes for Setting Configuration

Device Name

Command code: **BN**

Command parameters: The new device name for MiiNePort.

Reply parameters: N/A

?SBNMiiNePort@Office↵ !GBN0↵	System sets the device name as 'MiiNePort@Office'. MiiNePort reports command executed successfully.

Password

Command code: **BP**

Command parameters: MiiNePort's console password

Reply parameters: N/A

?SBP1234↵ !SBP0↵	System sets console password as '1234' for this MiiNePort. MiiNePort reports command executed successfully.

Web console

Command code: **BH**

Command parameters: 1 and web console TCP port to enable web console, or 0 to disable it.

Reply parameters: N/A

?SBH1;80↵ !SBH0↵	System sets http console as 'Enable' and http port as '80' for this MiiNePort. MiiNePort reports command executed successfully.

Telnet console

Command code: **BT**

Command parameters: 1 and telnet console TCP port if to enable telnet console, , or 0 to disable it.

Reply parameters: N/A

?SBT1;23↵ !SBT0↵	System sets telnet console as 'Enable' and telnet port as '23' for this MiiNePort. MiiNePort reports command executed successfully.

Auto configuration

Command code: **BA**

Command parameters: 1 if Auto configuration is enabled, otherwise 0

Reply parameters: N/A

?SBA1↵ !SBA0↵	System sets auto configuration as 'Enable' for this MiiNePort. MiiNePort reports command executed successfully.

External reset function

Command code: **BE**

Command parameters: 1 if external reset function is enabled, otherwise 0

Reply parameters: N/A

?SBE1↵ !SBE0↵	System sets external reset function as 'Enable' for this MiiNePort. MiiNePort reports command executed successfully.

NECI (utility accessibility)

Command code: **BU**

Command parameters: 1 if NECI is enabled, otherwise 0

Reply parameters: N/A

?SBU1↵ !SBU0↵	System sets NECI setting as 'Enable' for this MiiNePort. MiiNePort reports command executed successfully.

IP configuration

Command code: **NC**

Command parameters: MiiNePort's IP configuration index as shown in the following table

Reply parameters: N/A

0	Static
1	DHCP
2	DHCP/BOOTP
3	BOOTP
4	AutoIP

?SNC0↵ !SNC0↵	System sets IP configuration as 'Static'. MiiNePort reports command executed successfully.

IP address

Command code: **NI**

Command parameters: MiiNePort's IP address.

Reply parameters: N/A

?SNI192.168.1.2↵ !SNI0↵	System sets IP address as '192.168.1.2'. MiiNePort reports command executed successfully.

Netmask

Command code: **NM**

Command parameters: MiiNePort's Netmask address

Reply parameters: N/A

?SNM255.255.255.0↵ !SNM0↵	System sets Netmask as '255.255.255.0'. MiiNePort reports command executed successfully.

Gateway

Command code: **NG**

Command parameters: MiiNePort's Gateway address

Reply parameters: N/A

?SNG192.168.1.254↵ !SNG0↵	System sets Gateway as '192.168.1.254'. MiiNePort reports command executed successfully.

DNS

Command code: **ND**

Command parameters: The index (1,2) and DNS server address, separated by semicolon (;).

Reply parameters: N/A

?SND1;192.168.1.123↵ !SND0↵	System sets DNS1 as '192.168.1.123'. MiiNePort reports command executed successfully.

PHY Speed

Command code: **NS**

Command parameters: N/A

Reply parameters: MiiNePort's PHY speed as following:

0	PHY auto negotiation is enabled.
10	PHY speed is forced to 10Mbps with half duplex
11	PHY speed is forced to 10Mbps with full duplex
100	PHY speed is forced to 100Mbps with half duplex
101	PHY speed is forced to 100Mbps with full duplex

?SNS10↵ !SNS0↵	System forces PHY speed to 10Mbps with half duplex. MiiNePort reports command executed successfully.

Enable/Disable SNMP

Command code: **MS**

Command parameters: 1 (Enable) or 0 (Disable) MiiNePort's SNMP agent

Reply parameters: N/S

?SMS1↵ !SMS0↵	System sets SNMP agent as 'Enable'. MiiNePort reports command executed successfully.

Community String

Command code: **MU**

Command parameters: MiiNePort's SNMP Community string

Reply parameters: N/A

?SMUpublic↵ !SMU0↵	System sets SNMP community string as 'public'. MiiNePort reports command executed successfully.

Contact Name

Command code: **MN**

Command parameters: MiiNePort's SNMP contact name

Reply parameters: N/A

?SMNcontact↵ !SMN0↵	System sets SNMP contact name as 'contatct'. MiiNePort reports command executed successfully.

Location

Command code: **ML**

Command parameters: MiiNePort's SNMP location

Reply parameters: N/A

?SMLlocation↵ !SML0↵	System sets SNMP contact name as 'location'. MiiNePort reports command executed successfully.

Enable/Disable Accessible IP List

Command code: **AS**

Command parameters: 1 (Enable) or 0 (Disable) MiiNePort's accessible IP list

Reply parameters: N/A

?SAS1↵ !SAS0↵	System sets accessible IP list as 'Enable'. MiiNePort reports command executed successfully.

Accessible IP List

Command code: **AI**

Command parameters: format is "index;mode;IP;Netmask", where index ranges from 1 to 16, mode is 1 if activated and 0 if inactivated.

Reply parameters: N/A

?SAI1;1;192.168.1.2;255.255.255.0↵ !SAI0↵	System sets accessible IP 1 as 'active', IP address as '192.168.1.2' and netmask as '255.255.255.0'. MiiNePort reports command executed successfully.

Auto IP Report

Command code: **NR**

Command parameters: Auto IP report destination and report period.

Reply parameters: N/A

?SNR192.168.1.123:4000;50↵ !SNR0↵	System sets auto report IP as '192.168.1.123', UDP port as '4000' and report period as '50' sec. MiiNePort reports command executed successfully.

Port alias

Command code: **SA**

Command parameters: Port index and port alias, separated by semicolon (;).

Reply parameters: N/A

?SSA1;port 1↵ !SSA0↵	System sets port 1's alias name as 'port 1'. MiiNePort reports command executed successfully.

Baud rate

Command code: **SB**

Command parameters: Port index and baudrate separated by semicolon (;).

Reply parameters: N/A

?SSB1;115200↵ !SSB0↵	System sets port 1's baudrate as '115200'. MiiNePort reports command executed successfully.

Data bits

Command code: **SD**

Command parameters: Port index and data bits separated by semicolon (;).

Reply parameters: N/A

?SSD1;8↵ !SSD0↵	System sets port 1's data bits as '8'. MiiNePort reports command executed successfully.

Stop bits

Command code: **SS**

Command parameters: Port index and stop bits separated by semicolon (;). MiiNePort's Stop bits index as shown in the following table:

0	Stop bits is 1
1	Stop bits is 1.5
2	Stop bits is 2

Reply parameters: N/A

?SSS1;0↵ !SSS0↵	System sets port 1's stop bits as '1'. MiiNePort reports command executed successfully.

Parity

Command code: **SP**

Command parameters: Port index and parity separated by semicolon (;). MiiNePort's Parity index as shown in the following table:

0	None
1	Odd
2	Even
3	Mark
4	Space

Reply parameters: N/A

?SSP1;0↵ !SSP0↵	System sets port 1's parity bit as 'None'. MiiNePort reports command executed successfully.

Flow control

Command code: **SL**

Command parameters: Port index and flow control separated by semicolon (;).MiiNePort's Flow control index as shown in the following table:

0	None
1	RTS/CTS
2	XON/XOFF
3	DTR/DSR

Reply parameters: N/A

?SSL1;1↵ !SSL0↵	System sets port 1's flow control as 'RTS/CTS'. MiiNePort reports command executed successfully.

FIFO

Command code: **SF**

Command parameters: Port index and FIFO setting separated by semicolon (;). FIFO setting shall be 1 (Enable) or 0 (Disable).

Reply parameters: N/A.

?SSF1;1↵ !SSF0↵	System sets port 1's FIFO as 'Enable'. MiiNePort reports command executed successfully.

Operation Mode

Command code: **OM**

Command parameters: Port index, MCSC channel index, and operation mode to set. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. The operation mode is as following:

0	Disable
1	ReamCom
2	TCP
3	UDP
4	MCSC

Reply parameters: N/A

?SOM1;0;2↵ !SOM0↵	System sets port 1's operation mode as 'TCP'. MiiNePort reports command executed successfully.

Data Packing Length

Command code: **OL**

Command parameters: Port index, MCSC channel index, and data packing length to set. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?SOL1;0;256↵ !SOL0↵	System sets port 1's data packing length as '256'. MiiNePort reports command executed successfully.

Delimiter

Command code: **OD**

Command parameters: Port index, MCSC channel index, delimiter1 enable/disale, delimiter character 1 by hex, delimiter 2 enable/disable, and delimiter character 2 by hex. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?SOD1;2;1;10;1;13↵ !SOD0↵	System sets channel 2 of port 1's delimiter 1 as 'Enable' and hex code as '10'. Delimiter 2 as 'Enable' and hex code as '13'. MiiNePort reports command executed successfully.

Match Bytes

Command code: **OY**

Command parameters: Port index, MCSC channel index, and match bytes. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?SOY1;0;2↵ !SOY0↵	System sets port 1's match bytes as '2'. MiiNePort reports command executed successfully.

Delimiter Process

Command code: **OT**

Command parameters: N/A

Reply parameters: Port index, MCSC channel index, and delimiter process. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

?SOT1;0;2↵ !SOT0↵	System sets port 1's delimiter process as 'Delimiter+2'. MiiNePort reports command executed successfully.

Force Transmit

Command code: **OF**

Command parameters: Port index, MCSC channel index, and force transmit timeout. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?SOF1;0;5↵ !SOF0↵	System sets port 1's force transmit timeout as '5'. MiiNePort reports command executed successfully.

RealCOM TCP alive check time

Command code: **RA**

Command parameters: Port index, MCSC channel index, and TCP alive check time. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A.

?SRA1;0;5↵ !SRA0↵	System sets port 1's TCP alive check time as '5' for port 1. MiiNePort reports command executed successfully.

RealCOM max connection

Command code: **RM**

Command parameters: Port index, MCSC channel index, and maximum connection number. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?SRM1;1;4↵ !SRM0↵	System sets maximum connection number as '4' for port 1 and channel 1. MiiNePort reports command executed successfully.

RealCOM ignore jammed IP

Command code: **RJ**

Command parameters: Port index, MCSC channel index, and ignore jammed IP setting. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. Ignore jammed IP setting shall be 1 (Enable) or 0 (Disable).

Reply parameters: N/A

?SRJ1;2;0↵ !SRJ0↵	System sets ignore jammed IP policy as 'Disable' for port 1's channel 2. MiiNePort reports command executed successfully.

RealCOM allow Driver Control

Command code: **RD**

Command parameters: Port index, MCSC channel index, and match bytes. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. Allow driver control shall be 1 (Enable) or 0 (Disable).

Reply parameters: N/A

?SRD1;0;0↵ !SRD0↵	System sets allow driver control policy as 'Disable' for port 1. MiiNePort reports command executed successfully.

TCP Role

Command code: **TO**

Command parameters: Port index, MCSC channel index, and TCP role. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. TCP role shall be as following:

0	TCP server
1	TCP client
2	Mixed

Reply parameters: N/A

?STO1;0;1↵ !STO0↵	System sets port 1's TCP role to `TCP client'. MiiNePort reports command executed successfully.

TCP alive check time

Command code: **TA**

Command parameters: Port index, MCSC channel index, and TCP alive check time. Every parameter is separated by semicolon (;).

Reply parameters: N/A

?STA1;0;5↵ !STA0↵	System sets port 1's TCP alive check time to `5' minutes. MiiNePort reports command executed successfully.

TCP server connection control

Command code: **TS**

Command parameters: Port index, MCSC channel index, and server connection control setting. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. Server connection control setting shall be as following:

0	Always accept
1	Accept with DSR on

Reply parameters: N/A

?STS1;0;0↵ !STS0↵	System sets TCP server connection control as 'Always accept' for port 1. MiiNePort reports command executed successfully.

TCP password

Command code: **TW**

Command parameters: Port index, MCSC channel index, require (1) TCP connection password or not (0), and TCP connection password if required. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STW1;0;1;1234↵ !STW0↵	System sets password as 'required' and password as '1234' for port 1. MiiNePort reports command executed successfully.
?STW1;1;0↵ !STW0↵	System sets password is `not required' for port 1's channel 1. MiiNePort reports command executed successfully.

TCP communication protocol

Command code: **TR**

Command parameters: Port index, MCSC channel index, and communication protocol (0 for raw TCP and 1 for RFC-2217). Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STR1;0;0↵ !STR0↵	System sets communication protocol as 'Raw TCP' for port 1. MiiNePort reports command executed successfully.

TCP port

Command code: **TP**

Command parameters: Port index, MCSC channel index, and TCP local port. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STP1;0;4100↵ !STP0↵	System sets TCP port as '4100' for port 1. MiiNePort reports command executed successfully.

TCP client connection control

Command code: **TC**

Command parameters: Port index, MCSC channel index, and client connection control setting. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

0	Any character
1	Manual connection
2	Start up
3	With DSR on

?STC1;0;1↵ !STC0↵	System sets TCP client connection control as 'Manual connection' for port 1. MiiNePort reports command executed successfully.

TCP connection response

Command code: **TN**

Command parameters: Port index, MCSC channel index, and client connection response (0 for disable and 1 for enable). Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STN1;0;1↵ !STN0↵	System sets connection response as 'Enable' for port 1. MiiNePort reports command executed successfully.

TCP destination address

Command code: **TI**

Command parameters: Port index, MCSC channel index, destination address index (0 for destination address and 1 to 3 for alternated addresses). Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STI1;0;0;192.168.1.2:4001↵ !STT0↵	System sets destination address as '192.168.1.2' and port as '4001' for port 1. MiiNePort reports command executed successfully.

TCP connect retry timeout

Command code: **TT**

Command parameters: Port index, MCSC channel index, and client connection retry timeout. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STT1;0;1500↵ !STI0↵	System sets TCP connect retry timeout as '1500' for port 1. MiiNePort reports command executed successfully.

TCP max connection

Command code: **TM**

Command parameters: Port index, MCSC channel index, and maximum connection number. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STM1;1;3↵ !STM0↵	System sets maximum connection number as '3' for port 1 and channel 1. MiiNePort reports command executed successfully.

TCP ignore jammed IP

Command code: **RJ**

Command parameters: Port index, MCSC channel index, and ignore jammed IP setting (0 for disable and 1 for enable). Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STJ1;2;0↵ !STJ0↵	System sets ignore jammed IP policy as 'Disable' for port 1's channel 2. MiiNePort reports command executed successfully.

TCP disconnect with DSR off

Command code: **TL**

Command parameters: Port index, MCSC channel index, and disconnect with DSR off setting (0 for disable and 1 for enable). Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STL1;2;1↵ !STL0↵	System sets DSR off policy as 'Enable' for port 1's channel 2. MiiNePort reports command executed successfully.

TCP check EOT character

Command code: **TE**

Command parameters: Port index, MCSC channel index, enable EOT checking (1) or not (0), and EOT character by hex if enabled.. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STE1;0;1;04↵ !STE0↵	System sets check EOT character 'Enable' and EOT character as '0x04' for port 1. MiiNePort reports command executed successfully.
?STE1;0;0↵ !STE0↵	System disables EOT checking. MiiNePort reports command executed successfully.

TCP inactivity time

Command code: **TV**

Command parameters: Port index, MCSC channel index, and inactivity timeout setting. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?STV1;0;10↵ !STV0↵	System sets inactivity time as '10' ms for port 1. MiiNePort reports command executed successfully.

UDP serial format

Command code: **UT**

Command parameters: Port index, MCSC channel index, and UDP serial format. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. Serial format shall be as following:

0	Raw data
1	Packet command

Reply parameters: N/A

?SUT1;0;0↵ !SUT0↵	System sets UDP serial format as 'Raw data'for port 1. MiiNePort reports command executed successfully.

UDP destination address

Command code: **UD**

Command parameters: Port index, MCSC channel index, destination address index (1 to 4), and destination addresses. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0. The destination addresses shall be formatted as Begin address – End address: UDP port.

Reply parameters: N/A

?SUD1;0;3;192.168.1.3 -192.168.1.8:4001↵ !SUD0↵	System sets UDP destination address 3 is from '192.168.1.3' to '192.168.1.8' and port as '4001' for port 1. MiiNePort reports command executed successfully.

UDP local listen port

Command code: **UP**

Command parameters: Port index, MCSC channel index, and local UDP port. Every parameter is separated by semicolon (;). For MCSC-disabled port, channel index shall be 0.

Reply parameters: N/A

?SUP1;0;4001↲ !SUP0↲	System sets UDP local listen port as '4001' for port 1. MiiNePort reports command executed successfully.

Pin function

Command code: **PF**

Command parameters: PIN index 1 to 8, and PIN function, separated by semicolon (;). The PIN function shall be as following:

Value	Meaning	MiiNePort E1 constraint
5	Ready/RTS	Available to PIN 6 only.
6	CTS	Available to PIN 8 only.
7	DTR	Available to PIN 7 only.
8	DSR	Available to PIN 8 only.
9	485_en	Available to PIN 6 or PIN 7 only, and unable to set to both PINs.
10	DIO	
11	Reset to default	Available to PIN 7 only.

Note that for MiiNePort E1, only PIN 6 to 8 are configurable.

Reply parameters: N/A

?SPF6;9↲ !SPF0↲	System sets PIN 6's function as '485_en'. MiiNePort reports command executed successfully.

DIO initial mode

Command code: **PM**

Command parameters: DIO port index and initial mode (0 for input and 1 for output), separated by semicolon (;). For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO 0	PIN 6, can only be output
1	DIO 1	PIN 7
2	DIO 2	PIN 8, can only be input

Note that this configuration is valid only if related PIN function is set to DIO.

Reply parameters: N/A

?SPM0;1↵ !SPM0↵	System sets dio0 mode as 'output' for this MiiNePort. MiiNePort reports command executed successfully.

DIO initial state

Command code: **PS**

Command parameters: DIO port index and initial output state (0 for low and 1 for high).

Note that this configuration is valid only if DIO initial mode is set to output.

Reply parameters: N/A

?SPS0;1↵ !SPS0↵	System sets dio0 state as 'high' for this MiiNePort. MiiNePort reports command executed successfully.

Serial command mode trigger

Command code: **CT**

Command parameters: serial command mode trigger index as shown in the following table

Note that this configuration is valid only if port 1's operation mode is not set to MCSC.

Reply parameters: N/A

0	Disabe
1	H/W control pin(DIO1)
2	Activated by characters
3	Activated by break signal

?SCT2↵ !SCT0↵	System sets serial command mode as 'Activate by characters'. MiiNePort reports command executed successfully.

Serial command mode trigger characters

Command code: **CC**

Command parameters: serial command mode trigger characters, in hex, separated by semicolon (;).

Note that this configuration is valid only if serial command mode trigger method is set to `Activated by characters'.

Reply parameters: N/A

?SCC45;66;67↵ !SCC0↵	System sets serial command mode trigger characters as `ABC' (0x656667). MiiNePort reports command executed successfully.

Serial command mode trigger only at boot up

Command code: **CB**

Command parameters: 0 if serial command mode can be triggered at any time, or 1 if it can only be triggered at boot up.

Note that this configuration is valid only if serial command mode trigger method is set to `Activated by characters'.

Reply parameters: N/A

?SCB1↵ !SCB0↵	System sets serial command mode can only be triggered by characters at boot up. MiiNePort reports command executed successfully.

Gratuitous ARP

Command code: **VA**

Command parameters: 1 and send period if gratuitous ARP is enable; otherwise, 0.

Reply parameters: N/A

?SVA1;300↵ !SVA0↵	System sets gratuitous ARP as 'Enable' and send period as '300' sec for this MiiNePort. MiiNePort reports command executed successfully.

Command Codes for Retrieving Running Configuration

Device Name

Command code: **BN**

Command parameters: N/A

Reply parameters: MiiNePort's name.

?RBN↵ !RBN0MiiNePort_9527↵	System requests configured device name for this MiiNePort. MiiNePort reports device name as 'MiiNePort_9527'.

Console Password

Command code: **BP**

Command parameters: N/A

Reply parameters: MiiNePort's console password.

?RBP↵ !RBP01234↵	System requests console password for this MiiNePort. MiiNePort reports console password as '1234'.

Web console

Command code: **BH**

Command parameters: N/A

Reply parameters: 1 and web console TCP port separated by a semicolon (;) if web console is enabled, or a 0 if it is disabled.

?RBH↵ !RBH01;80↵	System requests http console setting for this MiiNePort. MiiNePort reports http console as 'Enable' and http port as '80'.

Telnet console

Command code: **BT**

Command parameters: N/A

Reply parameters: 1 and telnet console TCP port separated by a semicolon (;) if telnet console is enabled, or a 0 if it is disabled.

?RBT↵ !RBT01;23↵	System requests telnet console setting for this MiiNePort. MiiNePort reports telnet console as 'Enable' and telnet port as '23'.

Auto configuration

Command code: **BA**

Command parameters: N/A

Reply parameters: 1 if Auto configuration is enabled, otherwise reply 0.

?RBA↵ !RBA01↵	System requests Auto configuration setting for this MiiNePort. MiiNePort reports Auto configuration as 'Enable'.

External reset function

Command code: **BE**

Command parameters: N/A

Reply parameters: 1 if external reset function is enabled, otherwise 0.

?RBE↵ !RBE01↵	System requests external reset function setting for this MiiNePort. MiiNePort reports Auto configuration as 'Enable'.

NECI (utility accessibility)

Command code: **BU**

Command parameters: N/A

Reply parameters: 1 if NECI is enabled, otherwise 0.

?RBU↵ !RBU01↵	System requests NECI setting for this MiiNePort. MiiNePort reports NECI as 'Enable'.

IP configuration

Command code: **NC**

Command parameters: N/A

Reply parameters: MiiNePort's IP configuration index as following.

0	Static
1	DHCP
2	DHCP/BOOTP
3	BOOTP
4	AutoIP

?RNC↵ !RNC00↵	System requests IP configuration for this MiiNePort. MiiNePort reports IP configuration as 'Static'.

IP address

Command code: **NI**

Command parameters: N/A

Reply parameters: MiiNePort's IP address.

?RNI↵ !RNI0192.168.127.254↵	System requests IP address for this MiiNePort. MiiNePort reports IP address as '192.168.127.254'.

Netmask

Command code: **NM**

Command parameters: N/A

Reply parameters: MiiNePort's Netmask address.

?RNM↵ !RNM0255.255.255.0↵	System requests Netmask address for this MiiNePort. MiiNePort reports netmask as '255.255.255.0'.

Gateway

Command code: **NG**

Command parameters: N/A

Reply parameters: MiiNePort's Gateway address.

?RNG↵ !RNM0255.255.255.255↵	System requests Gateway address for this MiiNePort. MiiNePort reports Gateway address as '255.255.255.255'.

DNS

Command code: **ND**

Command parameters: The index (1 or 2) of DNS server.

Reply parameters: MiiNePort's DNS address.

?RND1↵ !RND0192.168.1.2↵	System requests DNS server 1 address for this MiiNePort. MiiNePort reports DNS server 1 address as '192.168.1.2'.

PHY Speed

Command code: **NS**

Command parameters: N/A

Reply parameters: MiiNePort's PHY speed as following:

0	PHY auto negotiation is enabled.
10	PHY speed is forced to 10Mbps with half duplex
11	PHY speed is forced to 10Mbps with full duplex
100	PHY speed is forced to 100Mbps with half duplex
101	PHY speed is forced to 100Mbps with full duplex

?RNS↵ !RNS00↵	System requests PHY speed setting for this MiiNePort. MiiNePort reports PHY speed is determined by PHY auto negotiation function.

Enable/Disable SNMP

Command code: **MS**

Command parameters: N/A

Reply parameters: 1 (Enable) or 0 (Disable) MiiNePort's SNMP agent.

?RMS↵ !RMS01↵	System requests Enable/Disable SNMP agent for this MiiNePort. MiiNePort reports SNMP agent as 'Enable'.

Community String

Command code: **MU**

Command parameters: N/A

Reply parameters: MiiNePort's SNMP Community string.

?RMU↵ !RMU0public↵	System requests SNMP community string for this MiiNePort. MiiNePort reports SNMP community string as 'public'.

Contact Name

Command code: **MN**

Command parameters: N/A

Reply parameters: MiiNePort's SNMP Contact Name.

?RMN↵ !RMN0s_name↵	System requests SNMP contact name for this MiiNePort. MiiNePort reports SNMP contact name as 's_name'.

Location

Command code: **ML**

Command parameters: N/A

Reply parameters: MiiNePort's SNMP Location.

?RML↵ !RML0s_location↵	System requests SNMP location for this MiiNePort. MiiNePort reports SNMP location as 's_location'.

Enable/Disable Accessible IP List

Command code: **AS**

Command parameters: N/A

Reply parameters: Enable (1) or Disable (0) MiiNePort's accessible IP list.

?RNS↵ !RNS01↵	System requests Enable/Disable accessible IP list for this MiiNePort. MiiNePort reports accessible IP list as 'Enable'.

Accessible IP List

Command code: **AI**

Command parameters: The index of accessible IP list (from 1 to 16)

Reply parameters: Return format Mode;IP;Netmask in the accessible IP list. If mode equals to 1, it's active, otherwise it's inactive.

?RAI1↵ !RAI01;192.168.1.2;255.255.255.0↵	System requests 1st accessible IP list for this MiiNePort. MiiNePort reports 1st assessible IP list as 'Active', IP address as '192.167.1.2', and Netmask as '255.255.255.0' .

Auto IP Report

Command code: **NR**

Command parameters: N/A

Reply parameters: MiiNePort's auto IP report setting.

?RNR↵ !RNR0192.168.1.250:4000;50↵	System requests auto IP report for this MiiNePort. MiiNePort reports Auto IP report server as '192.168.1.250' and port as '4000', and report period as '50'.

Port alias

Command code: **SA**

Command parameters: N/A

Reply parameters: MiiNePort's Port alias.

?RSA↵ !RSA0port1↵	System requests Port alias for this MiiNePort. MiiNePort reports Port alias as 'port1'.

Baud rate

Command code: **SB**

Command parameters: Port index.

Reply parameters: MiiNePort's Baud rate.

?RSB1↵ !RSB0115200↵	System requests port 1's Baud rate for this MiiNePort. MiiNePort reports Baud rate as '115200'.

Data bits

Command code: **SD**

Command parameters: Port index.

Reply parameters: MiiNePort's Data bits.

?RSD1↵ !RSD08↵	System requests port 1's Data bits for this MiiNePort. MiiNePort reports Data bits as '8'.

Stop bits

Command code: **SS**

Command parameters: Port index.

Reply parameters: MiiNePort's stop bits index.

Stop Bits Index Table

0	Stop bits is 1
1	Stop bits is 1.5
2	Stop bits is 2

?RSS1↵ !RSS00↵	System requests port 1's Stop bits for this MiiNePort. MiiNePort reports Stop bits as '1'.

Parity

Command code: **SP**

Command parameters: Port index

Reply parameters: MiiNePort's Parity index as following.

0	None
1	Odd
2	Even
3	Mark
4	Space

?RSP1↵ !RSP00↵	System requests port 1's Parity for this MiiNePort. MiiNePort reports Parity as 'None'.

Flow control

Command code: **SL**

Command parameters: Port index.

Reply parameters: MiiNePort's Flow control index as following.

0	None
1	RTS/CTS
2	XON/XOFF
3	DTR/DSR

?RSL1↵ !RSL01↵	System requests port 1's Flow control for this MiiNePort. MiiNePort reports Flow control as 'RTS/CTS'.

FIFO

Command code: **SF**

Command parameters: Port index.

Reply parameters: 1 (Enable) or 0 (Disable) MiiNePort's FIFO.

?RSF1↵ !RSF01↵	System requests port 1's FIFO for this MiiNePort. MiiNePort reports FIFO as 'Enable'.

Operation Mode

Command code: **OM**

Command parameters: numbers separated by a semicolon (;) denotes port index and MCSC channel index. For port itself, channel index shall be 0.

Reply parameters: MiiNePort's operation mode index as shown in the following table.

0	Disable
1	RealCom
2	TCP
3	UDP
4	MCSC (This value is unavailable for MCSC channels.)

?ROM1;0↵ !ROM04↵ ?ROM1;2↵ !GOM02↵	System requests port 1's Operation Mode for this MiiNePort. MiiNePort reports Flow control as 'MCSC'. When port 1 is set to MCSC, system requests operation mode for channel 2 of port 1. MiiNePort reports operation mode is `TCP'.

Data Packing Length

Command code: **OL**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's data packing length as following.

?ROL1;0↵ !ROL0256↵	System requests port 1's data packing length for this MiiNePort. MiiNePort reports data packing length as '256'.

Delimiter

Command code: **OD**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's delimiter setting. (delimiter1 enable/disable;hex1;delimiter2 enable/disable;hex2)

?ROD1;0↵ !ROD1;10;1;13↵	System requests port 1's delimiter setting for this MiiNePort. MiiNePort reports delimiter 1 as 'Enable' and hex code as '10', delimiter 2 as 'Enable' and hex code as '13'

Match Bytes

Command code: **OY**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's match bytes.

?ROY1;0↵ !ROY02↵	System requests port 1's match bytes for this MiiNePort. MiiNePort reports match bytes as '2'.

Delimiter Process

Command code: **OT**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's delimiter process.

0	Do nothing
1	Delimiter+1
2	Delimiter+2

?ROT1;0↵ !ROT02↵	System requests port 1's delimiter process for this MiiNePort. MiiNePort reports delimiter process as 'Delimiter+2'.

Force Transmit

Command code: **OF**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's force transmit timeout.

?ROF1;0↵ !ROF00↵	System requests port 1's force transmit timeout for this MiiNePort. MiiNePort reports force transmit timeout as '0' sec.

RealCOM TCP alive check time

Command code: **RA**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's TCP alive check time.

?RRA1;0↵ !RRA05↵	System requests port 1's TCP check alive time for port 1. MiiNePort reports TCP alive check time as '5' min.

RealCOM max connection

Command code: **RM**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: MiiNePort's TCP maximum connection number.

?RRM1;1↵ !RRM03↵	System requests maximum connection number for port 1 and channel 1. MiiNePort reports maximum connection number as '3'.

RealCOM ignore jammed IP

Command code: **RJ**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?RRJ1;2↵ !RRJ01↵	System requests jammed IP policy for port 1's channel 2. MiiNePort reports ignore jammed IP as 'Enable'.

RealCOM Allow Driver Control

Command code: **RD**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?RRD1;0↵ !RRD01↵	System requests allow driver control policy for port1. MiiNePort reports allow driver control as 'Enable'.

TCP Role

Command code: **TO**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP role as shown in the following table.

0	TCP server
1	TCP client
2	Mixed

?RTO1;0↵	System requests TCP role for port 1.
!RTO00↵	MiiNePort reports TCP role is TCP server.

TCP server connection control

Command code: **TS**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP server connection control as shown in the following table.

0	Always accept
1	Accept with DSR on

?RTS1;0↵	System requests accept incoming policy for port 1.
!RTS00↵	MiiNePort reports incoming policy as 'Always accept'.

TCP password

Command code: **TW**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Require) or 0 (Don't require) and password.

?RTW1;0↵	System requests port 1's password setting for port 1.
!RTW01;1234↵	MiiNePort reports password as 'required' and password as '1234'.

TCP communication protocol

Command code: **TR**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 0 (Raw TCP) or 1 (RFC 2217)

?RTR1;0↵ !RTR00↵	System requests communication protocol for port 1. MiiNePort reports communication protocol as 'Raw TCP'.

TCP alive check time

Command code: **TA**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP alive check time.

?RTA1;0↵ !RTA05↵	System requests TCP alive check time for port 1. MiiNePort reports TCP alive check time as '5' minutes.

TCP port

Command code: **TP**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP port.

?RTP1;0↵ !RTP04100↵	System requests TCP port for port 1. MiiNePort reports TCP port as '4100'.

TCP client connection control

Command code: **TC**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP client connection control index as shown in the following table.

0	Any character
1	Manual connection
2	Start up
3	With DSR on

?RTC1;0↵ !RTC01↵	System requests TCP client connection control for port 1. MiiNePort reports TCP client connection control as 'Any character'.

TCP connection response

Command code: **TN**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable).

?RTN1;0↵ !RTN00↵	System requests connection response policy for port 1. MiiNePort reports connection response as 'Disable'.

TCP destination address

Command code: **TI**

Command parameters: Three numbers separated by a semicolon (;) denotes port index, MCSC channel index, and destination address index. For MCSC-disabled port, channel index shall be 0. The destination address index is as following:

0	Destination address
1	Alternated address 1
2	Alternated address 2
3	Alternated address 3

Reply parameters: TCP destination address and port

?RTI1;0;1↵ !RTI0192.168.1.2:4001↵	System requests destination address 1 for port 1. MiiNePort reports destination address as '192.168.1.2' and port as '4001'.

TCP connect retry timeout

Command code: **TT**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP connect retry timeout.

?RTT1;0↵ !RTI01500↵	System requests TCP connect retry timeout for port 1. MiiNePort reports TCP connect retry timeout as '1500'.

TCP max connection

Command code: **TM**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

?RTM1;1↵ !RTM04↵	System requests maximum connection number for port 1 and channel 1. MiiNePort reports maximum connection number as '4'.

TCP ignore jammed IP

Command code: **TJ**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable).

?RTJ1;2↵ !RTJ01↵	System requests jammed IP policy for port 1's channel 2. MiiNePort reports ignore jammed IP as 'Enable'.

TCP disconnect whit DSR off

Command code: **TL**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable).

?RTL1;2↵ !RTL01↵	System requests DSR off policy for port 1's channel 2. MiiNePort reports on DSR off as 'Enable'.

TCP check EOT character

Command code: **TE**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: 1 (Enable) or 0 (Disable), and EOT character.

?RTE1;0↵ !RTE01;04↵	System requests check EOT character setting for port 1. MiiNePort reports check EOT character as 'Enable' and EOT character as '0x04'.

TCP inactivity time

Command code: **TV**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: TCP inactivity time.

?RTV1;0↵ !RTV00↵	System requests inactivity time for port 1. MiiNePort reports inactivity time as '0'.

UDP serial format

Command code: **UT**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: Serial format as following:

0	Raw data
1	Packet command

?RUT1;0↲ !RUT00↲	System requests UDP serial format for port 1. MiiNePort reports UDP serial format as 'Raw data'.

UDP destination address

Command code: **UD**

Command parameters: Three numbers separated by a semicolon (;) denotes port index, MCSC channel index, and destination address index from 1 to 4. For MCSC-disabled port, channel index shall be 0.

Reply parameters: Begin address – End address: port

?RUD1;0;3↲ !RUD0192.168.1.3-192.168.1. 8:4001↲	System requests UDP destination address 3 for port 1. MiiNePort reports UDP destination address is from '192.168.1.3' to '192.168.1.8' and port as '4001'.

UDP local listen port

Command code: **UP**

Command parameters: Two numbers separated by a semicolon (;) denotes port index and MCSC channel index. For MCSC-disabled port, channel index shall be 0.

Reply parameters: local listen port.

?RUP1;0↲ !RUP4001↲	System requests UDP local listen port for port 1. MiiNePort reports UDP local listen port as '4001'.

Pin function

Command code: **PF**

Command parameters: PIN index from 1 to 8.

Reply parameters: The PIN function is as following:

0	GND
1	VCC
2	RST
3	Tx
4	Rx
5	Ready/RTS
6	CTS
7	DTR
8	DSR
9	485_en
10	DIO
11	Reset to default

?RPF1↵	System requests PIN 1's function for this MiiNePort.
!RPF00↵	MiiNePort reports PIN function as 'GND'.
?GPF7↵	System requests PIN 7's function for MiiNePort.
!GPF011↵	MiiNePort reports PIN function is `Reset to default'.

DIO initial mode

Command code: **PM**

Command parameters: DIO port index. For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Reply parameters: 1 (output) or 0 (input) , note that for MiiNePort E1, this configuration is available if specific PIN function is set to DIO.

?RPM1↵	System requests DIO1's initial mode for this MiiNePort.
!RPM01↵	MiiNePort reports DIO1's initial mode as 'output'.

DIO initial state

Command code: **PS**

Command parameters: DIO port index. For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Reply parameters: 1 (high) or 0 (low) , note that this function is configuration only if specific DIO port is set to initial output.

?RPS0↵ !RPS00↵	System requests DIO0's initial state for this MiiNePort. MiiNePort reports DIO0's initial state as 'low'.

Serial command mode trigger

Command code: **CT**

Command parameters: N/A

Reply parameters: serial command mode trigger index as shown in the following table.

0	Disabe
1	H/W control pin(DIO1)
2	Activated by characters
3	Activated by break signal

Note that this configuration is valid only if port 1's operation mode is not set to MCSC.

?RCT↵ !RCT01↵	System requests serial command mode trigger method for this MiiNePort. MiiNePort reports serial command mode is triggered by DIO1.

Serial command mode trigger characters

Command code: **CC**

Command parameters: N/A

Reply parameters: serial command mode trigger characters, in hex, separated by semicolon (;).

Note that this configuration is valid only if serial command mode trigger method is set to `Activated by characters'.

?RCC↵ !RCC058;58;58↵	System requests serial command mode trigger characters for this MiiNePort. MiiNePort reports serial command mode triggered by `XXX' (0x585858).

Serial command mode trigger only at boot up

Command code: **CB**

Command parameters: N/A

Reply parameters: 0 if serial command mode can be triggered at any time, or 1 if it can only be triggered at boot up.

Note that this configuration is valid only if serial command mode trigger method is set to `Activated by characters'.

?RCB↵ !RCB01↵	System requests serial command mode for this MiiNePort. MiiNePort reports serial command mode can only be triggered by characters at boot up.

Gratuitous ARP

Command code: **VA**

Command parameters: N/A.

Reply parameters: 1 and send period if gratuitous ARP is enable; otherwise, reply 0.

?RVA↵ !RVA1;300↵	System requests gratuitous ARP status for alert mail server. MiiNePort reports gratuitous ARP as 'Enable' and send period as '300' sec.

Serial Number

Command code: **@S**

Command parameters: N/A

Reply parameters: MiiNePort's serial number, in digit notation.

?R@S↵ !R@S09527↵	System requests MiiNePort's serial number. MiiNePort reports serial number is 9527.

Firmware Version

Command code: **@V**

Command parameters: N/A

Reply parameters: MiiNePort's firmware version. Version parts are separated by single dot, with at least two parts (major.minor) and at most three parts (major.minor.rev).

?R@V↵ !R@S01.0↵	System requests MiiNePort's firmware version. MiiNePort reports serial number is 9527.

Firmware Build Number

Command code: **@B**

Command parameters: N/A

Reply parameters: MiiNePort's firmware build number, eight digits.

?R@B↵ !R@B009021208↵	System requests MiiNePort's firmware build number. MiiNePort reports build number is 09021208.

MAC address

Command code: **@M**

Command parameters: N/A

Reply parameters: MiiNePort's MAC address.

?R@M↵ !R@M000:90:E8:71:00:10↵	System requests MiiNePort's firmware build number. MiiNePort reports MAC address is 00:90:E8:71:00:10.

Command Codes for Viewing Status

Ethernet Speed and Duplex

Command code: **NS**

Command parameters: N/A

Reply parameters: The speed and duplex status of Ethernet. Possible values are 0(Unlink), 10H(10Mbps Half Duplex), 10F(10Mbps Full Duplex), 100H(100Mbps Half Duplex), 100F(100Mbps Full Duplex).

?VNS↵ !VNS0100F↵	System requests current Ethernet status. MiiNePort reports the Ethernet is current in 100Mbps Full Duplex mode.

DIO mode

Command code: **PM**

Command parameters: DIO port index and IO mode (0 for input and 1 for output), separated by semicolon (;). For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO 0	PIN 6, can only be output
1	DIO 1	PIN 7
2	DIO 2	PIN 8, can only be input

Note that this function works only if related PIN function is set to DIO.

Reply parameters: N/A

?CPM1;0↵ !CPM0↵	System sets DIO 1's mode as 'input' for this MiiNePort. MiiNePort reports command executed successfully.

DIO state

Command code: **PS**

Command parameters: DIO port index and output state (0 for low and 1 for high).

Note that this function works only if DIO mode is set to output.

Reply parameters: N/A

?CPS0;1↵ !CPS0↵	System sets DIO 0's state to 'high'. MiiNePort reports command executed successfully.

Command Codes for Controlling

PING

Command code: **NP**

Command parameters: Target host name or IP address.

Reply parameters: A single minus symbol indicates the target host did not reply in 1000 milliseconds. Otherwise, one decimal number indicating the reply latency in milliseconds is returned.

?CNPwww.moxa.com↵ !CNP010↵	System requests to PING www.moxa.com. Target host replies in 10 milliseconds.
?CNP192.168.1.1↵ !CNP0-↵	System requests to PING 192.168.1.1. Target host did not reply in 1000 milliseconds.

DIO mode

Command code: **PM**

Command parameters: DIO port index. For MiiNePort E1, the index is as following:

Index	DIO port	MiiNePort E1 PIN number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Reply parameters: 1 (output) or 0 (input), note that for MiiNePort E1, this configuration is available if specific PIN function is set to DIO.

?GPM1↵ !GPM01↵	System requests DIO1's initial mode for this MiiNePort. MiiNePort reports DIO1's initial mode as 'output'.

DIO state

Command code: **PS**

Command parameters: DIO port index.

Reply parameters: If DIO port is in output mode, a 0 denotes it is outputting low signal, or 1 for high signal. If DIO port is in input mode, a 0 denotes it receives low signal, and 1 for high signal.

?VPS2↵	System requests DIO state of DIO port 2.
!VPS01↵	MiiNePort reports DIO port is in high state.

Save & Restart

Command code: **SR**

Command parameters:'1' for "Save & Restart", '0' for "Restart only".

Reply parameters: N/A

?CSR1↵	System requests to save configuration and restart.
!CSR0	Your configuration is saved, and restart MiiNePort server now.

Load Factory Default

Command code: **LD**

Note that this command do not apply configuration until you save & restart MiiNePort.

Command parameters: '1' for "All setting", '0' for "Keep IP setting".

Reply parameters: N/A

?CLD0↵	System requests to load factory default.
!CLD0↵	System loads factory default successfully.

B

Well Known Port Numbers

This appendix is included for your reference. Listed below are port numbers that already have a well-established use. These port numbers should be avoided when assigning a port number to your MiiNePort E1 Series module; otherwise you may experience network problems. Refer to the RFC 1700 standard for Well Known Port Numbers or refer to the following introduction from IANA.

The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports.

The Well Known Ports are those from 0 through 1023.

The Registered Ports are those from 1024 through 49151.

The Dynamic and/or Private Ports are those from 49152 through 65535.

The Well Known Ports are assigned by IANA, and on most systems, can only be used by system processes or by programs executed by privileged users. The following table shows famous port numbers among the well-known port numbers. For more details, please visit the IANA website at http://www.iana.org/assignments/port-numbers.

TCP Socket	Application Service
0	reserved
1	TCP Port Service Multiplexor
2	Management Utility
7	Echo
9	Discard
11	Active Users (systat)
13	Daytime
15	Netstat
20	FTP data port
21	FTP CONTROL port
23	Telnet
25	SMTP (Simple Mail Transfer Protocol)
37	Time (Time Server)
42	Host name server (names server)
43	Whois (nickname)
49	(Login Host Protocol) (Login)
53	Domain Name Server (domain)
79	Finger protocol (Finger)

TCP Socket	Application Service
80	World Wide Web HTTP
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol
213	IPX
160 – 223	Reserved for future use

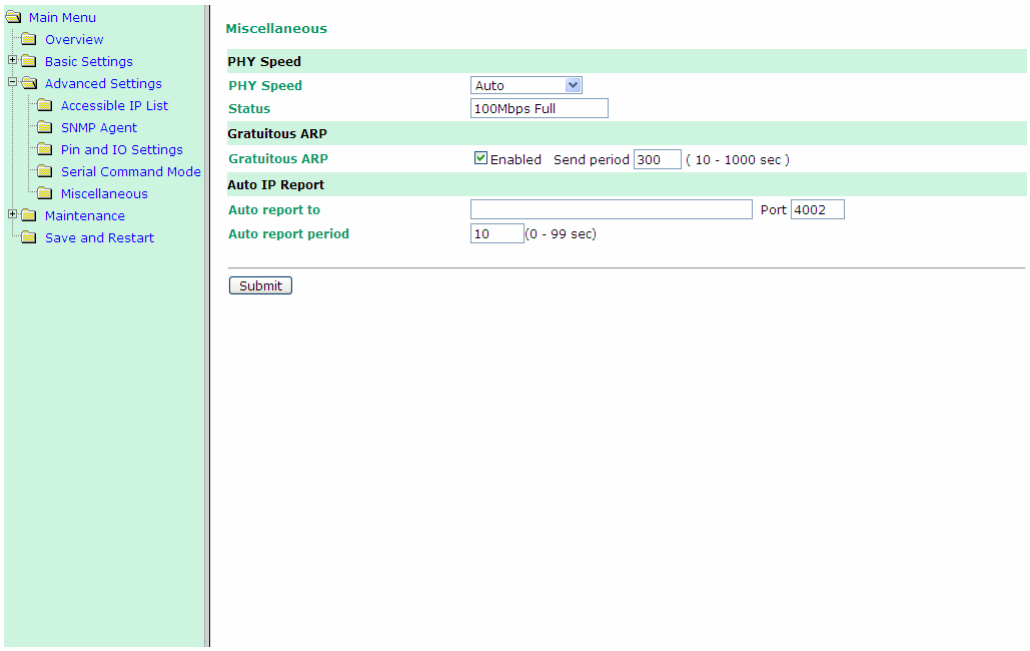
UDP Socket	Application Service
0	reserved
2	Management Utility
7	Echo
9	Discard
11	Active Users (systat)
13	Daytime
35	Any private printer server
39	Resource Location Protocol
42	Host name server (names server)
43	Whois (nickname)
49	(Login Host Protocol) (Login)
53	Domain Name Server (domain)
69	Trivial Transfer Protocol (TETP)
70	Gopler Protocol
79	Finger Protocol
80	World Wide Web HTTP
107	Remote Telnet Service
111	Sun Remote Procedure Call (Sunrpc)
119	Network news Tcanster Protocol (NNTP)
123	Network Time protocol (nnp)
161	SNMP (Simple Network Mail Protocol)
162	SNMP Traps
213	IPX (Used for IP Tunneling)

C

Auto IP Report Protocol

There are several ways to configure the IP address of an MiiNePort E1 Series module. One of them is DHCP Client. When you set up the module to use DHCP Client for IP address configuration, it will automatically send a DHCP request over the network to find the DHCP server. The DHCP server will then send an available IP address to the module with an expiration time. The module will use this IP address until the expiration time has been reached. When the expiration time has been reached, the process will repeat, and module will send another DHCP request to the DHCP server. Therefore, a module may end up using more than one IP address while it is connected to the network.

The module has a built-in IP address report function that will send its IP data to a specific IP address and port number when it is not using a Static or fixed IP address. The IP address report parameters may be configured in the Network Settings in the web console. Enter the IP address and the port number of the PC that is to receive this information as shown below:



IP Address Report Structure

The first 4 bytes of the module's IP address report are the characters "MOXA". The rest of the report is composed of 9 items, with each item preceded by a 2-byte header indicating the item ID and item length.

Header (Item ID)	Header, (Item Length)	Item
(none)	(none)	"Moxa" (text string)
1	(varies)	server name (text string)
2	2	hardware ID (little endian, see table below)
3	6	MAC address (00-90-E8-01-02-03 would be sent in sequence as 0x00, 0x90, 0xE8, 0x01, 0x02, 0x03)
4	4	serial number (little endian DWORD)
5	4	IP address
6	4	netmask
7	4	default gateway
8	4	firmware version (little endian DWORD, Version 4.3.1= 0x04030100)
9	4	AP ID (little endian DWORD, see table below)

Example

The following example shows the first 22 bytes of a typical IP address report:

	report header "Moxa"				item ID	item length	server name "TEST"				item ID	item length	hardware ID 0x4119		item ID	item length	MAC address 00-90-E8-01-02-03					
HEX	4D	4F	58	41	01	04	54	45	53	54	02	02	19	41	03	06	00	90	E8	01	02	03
ASCII	"M"	"O"	"X"	"A"			"T"	"E"	"S"	"T"												

D

DIO Commands

In this appendix, we provide information on sending commands to the module's DIO channels over an Ethernet network. Digital I/O commands and responses are accessed using a specific TCP port (default 5001) on the module. Each command is initiated by the host and is followed by a response from the module. A utility on the CD-ROM can be used to test the DIO access commands.

Overview

Each DIO command and response consists of a 4-byte header and up to 255 bytes of data. The first byte of the header indicates the command. The second byte indicates the version, which is "2" for current firmware versions. The third byte is a code that is used by the module to report errors. The fourth byte is the number of bytes that follows the header, and will depend on the command or response.

In the event of an error, the module will return the entire command as its response, but with the third byte changed according to the following status/error codes:

1: Command error; may be unknown

2: Version error; not supported by this version

3: Length error; the length member does not match the attached data

4: Operation error; invalid status or invalid mode

5: "Packet too short" error

6: DIO number error; might not support request DIO number

0xFF: other unknown error

ATTENTION

DIO command data is transmitted as values rather than text strings. A value of 1 would be transmitted as 0x01.

C Code Example

```
//define DIO Header format
typedef struct _DIO_Header_Struct {
char command;
char version; /* This specification is version 2 */
char status;
char length;
} DIOHeaderStruct, *pDIOHeaderStruct;
//define DIO Packet format
//Used for Command and ACK packet
typedef struct _DIO_Packet_Struct {
DIOHeaderStruct header;
Char data[ 255];
} DIOPacketStruct, *pDIOPacketStruct;
```

Read Single DIO

Command

Byte #	Descriptor	Value	Description
1	Header	1	command #, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module's response
4	Header	1	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel #

For example, the 5-byte command sequence **1-2-0-1-0** requests the status of DIO 0.

Response

Byte #	Descriptor	Value	Description
1	Header	1	command #, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	3	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel #
6	Data	0, 1	0: channel is in input mode 1: channel is in output mode
7	Data	0, 1	0: channel status is low 1: channel status is high

For example, the 7-byte response sequence **1-2-0-3-0-0-0** indicates that DIO 0 is in input mode.

C Code Example

```
BOOL ReadSingleDIO(int port, int *mode, int *status)
{
DIOPacketStruct packet;
packet.header.command = 1; // read single DIO command
packet.header.version = 2; // DIO protocol version
packet.header.length = 1; // data length
packet.data[0] = (char)port; // Number of the DIO
send(SocketFd, (char *)&packet, sizeof(DIOHeaderStruct)+1, 0);
//Send TCP Packet
// Process the returned data here.
return TRUE;
}
```

Write Single DIO

Command

Byte #	Descriptor	Value	Description
1	Header	2	command number, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module's response
4	Header	3	data length, fixed
5	Data	0,1,2,3	desired DIO channel number
6	Data	0, 1	0: set to input mode 1: set to output mode
7	Data	0, 1	this byte is ignored for input mode 0: set to low 1: set to high

For example, the 7-byte command sequence **2-2-0-3-0-0-0** requests that DIO 0 be set to digital input mode.

Response

Byte #	Descriptor	Value	Description
1	Header	2	command number, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	3	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel #
6	Data	0, 1	0: channel has been changed to input mode 1: channel has been changed to output mode
7	Data	0, 1	this byte is ignored for input mode 0: channel status has been changed to low 1: channel status has been changed to high

For example, the 7-byte response sequence **2-2-0-3-0-0-0** indicates that DIO 0 has been changed to input mode.

C Code Example:

```
void WriteSingleDIO(int port, int mode, int status)
{
DIOPacketStruct packet;
packet.header.command = 2; // write single DIO command
packet.header.version = 2; // DIO protocol version
packet.header.length = 3; // data length
packet.data[0] = (char)port; // number of the DIO
packet.data[1] = (char)mode; // DIO mode
packet.data[2] = (char)status; // DIO status;
send(SocketFd, (char *)&packet, sizeof(DIOHeaderStruct)+3, 0);
//Send TCP packet
//Process the returned data here
}
```

Read Multiple DIO

Command

Byte #	Descriptor	Value	Description
1	Header	5	command number, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module's response
4	Header	2	data length, fixed
5	Data	0,1,2	starting DIO channel number
6	Data	1,2,3	ending DIO channel number

This command requests the status of a range of DIO channels, specified in bytes 5 and 6. For example, the 6-byte command sequence 5-2-0-2-0-2 requests the status of DIO channels 0 through 2.

Response

Byte #	Descriptor	Value	Description
1	Header	5	command number, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	4, 6, 8	data length, depends on the number of DIO channels requested
5	Data	0, 1	1st requested DIO channel 0: channel is in input mode 1: channel is in output mode
6	Data	0, 1	1st requested DIO channel 0: channel status is low 1: channel status is high
7	Data	0, 1	2nd requested DIO channel 0: channel is in input mode 1: channel is in output mode

Byte #	Descriptor	Value	Description
8	Data	0, 1	2nd requested DIO channel 0: channel status is low 1: channel status is high
9	Data	0, 1	3rd requested DIO channel, optional 0: channel is in input mode 1: channel is in output mode
10	Data	0, 1	3rd requested DIO channel, optional 0: channel status is low 1: channel status is high
11	Data	0, 1	4th requested DIO channel, optional 0: channel is in input mode 1: channel is in output mode
12	Data	0, 1	4th requested DIO channel, optional 0: channel status is low 1: channel status is high

For example, the 10-byte response **5-2-0-6-0-0-1-1-0-1** indicates that DIO 0 is in input mode and "low" status, DIO 1 is in output mode and "high" status, and DIO 2 is in input mode and "high" status.

C Code Example:

```
BOOL ReadMultipleDIO(int start, int end, int *mode, int *status)
{
DIOPacketStruct packet;
packet.header.command = 5; // Read Multiple DIO Commands
packet.header.version = 2; // DIO protocol command version
packet.header.length = 2; // data length
packet.data[0] = start; // start of the DIO number
packet.data[1] = end; // end of the DIO number
send(SocketFd, (char *)&packet, sizeof(DIOHeaderStruct)+2, 0);
//Send TCP packet
//Process the returned data here
return TRUE;
}
```

Write Multiple DIO

Command

Byte #	Descriptor	Value	Description
1	Header	6	command number, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module's response
4	Header	6, 8, 10	data length, depends on the number of channels being written (6 bytes for 2 channels, 8 bytes for 3 channels, 10 bytes for 4 channels)
5	Data	0,1,2	starting DIO channel number
6	Data	1,2,3	ending DIO channel number

Byte #	Descriptor	Value	Description
7	Data	0,1	1st DIO channel to be written 0: set to input mode 1: set to output mode
8	Data	0,1	1st DIO channel to be written 0: set to low 1: set to high
9	Data	0,1	2nd DIO channel to be written 0: set to input mode 1: set to output mode
10	Data	0,1	2nd DIO channel to be written 0: set to low 1: set to high
11	Data	0,1	3rd DIO channel to be written, optional 0: set to input mode 1: set to output mode
12	Data	0,1	3rd DIO channel to be written, optional 0: set to low 1: set to high
13	Data	0,1	4th DIO channel to be written, optional 0: set to input mode 1: set to output mode
14	Data	0,1	4th DIO channel to be written, optional 0: set to low 1: set to high

This command writes the status of a range of DIO channels, specified in bytes 5 and 6. The length of the command depends on the number of channels to be written. For example, the 10-byte command **6-2-0-6-0-1-0-0-1-1** requests DIO 0 be set to digital input mode and "low" status and DIO 1 be set to digital output mode and "high" status. If you wanted to include a change of DIO 2 to digital output mode and "low" status, the 12-bye command sequence would be **6-2-0-8-0-2-0-0-1-1-1-0**.

Response

Byte #	Descriptor	Value	Description
1	Header	6	command number, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	4, 6, 8	data length, depends on the number of DIO channels requested
5	Data	0, 1	1st requested DIO channel 0: channel has been changed to input mode 1: channel has been changed to output mode
6	Data	0, 1	1st requested DIO channel 0: channel status has been changed to low 1: channel status has been changed to high

Byte #	Descriptor	Value	Description
7	Data	0, 1	2nd requested DIO channel 0: channel has been changed to input mode 1: channel has been changed to output mode
8	Data	0, 1	2nd requested DIO channel 0: channel status has been changed to low 1: channel status has been changed to high
9	Data	0, 1	3rd requested DIO channel, optional 0: channel has been changed to input mode 1: channel has been changed to output mode
10	Data	0, 1	3rd requested DIO channel, optional 0: channel status has been changed to low 1: channel status has been changed to high
11	Data	0, 1	4th requested DIO channel, optional 0: channel has been changed to input mode 1: channel has been changed to output mode
12	Data	0, 1	4th requested DIO channel, optional 0: channel status has been changed to low 1: channel status has been changed to high

For example, the 8-byte response 6-2-0-4-0-0-1-1 indicates that DIO 0 has been changed to input mode and "low" status and DIO 1 has been changed to output mode and "high" status.

C Code Example:

```
void WriteMultipleDIO(int start, int end, int* mode, int* status)
{
DIOPacketStruct packet;
packet.header.command = 6; // Write Multiple DIO Command Codes
packet.header.version = 2; // DIO protocol version
packet.header.length = (end-start+1)*2+2; // data length
packet.data[0] = start; // start DIO number
packet.data[1] = end; // end DIO number
int i, len;
for ( i=0; i<(end-start+1);i++ ) {
packet.data[i+2] = mode[i];
packet.data[i+3] = status[i];
}
send(SocketFd, )(char*)&packet,(end-start+1)*2+2+sizeof(DIOHeaderStruct), 0);
//Send TCP packet
//Process the returned data here
}
```

E

SNMP Agent with MIB II & RS-232 Like Group

MiiNePort E1 Series modules have built-in SNMP (Simple Network Management Protocol) agent software. It supports RFC1317 RS-232 like group and RFC 1213 MIB-II. The following table lists the standard MIB-II groups, as well as the variable implementations for MiiNePort E1 Series modules.

RFC1 213 MIB-II supported SNMP variables:

System MIB	Interfaces MIB	IP MIB	ICMP MIB
SysDescr	itNumber	ipForwarding	IcmpInMsgs
SysObjectID	ifIndex	ipDefaultTTL	IcmpInErrors
SysUpTime	ifDescr	ipInreceives	IcmpInDestUnreachs
SysContact	ifType	ipInHdrErrors	IcmpInTimeExcds
SysName	ifMtu	ipInAddrErrors	IcmpInParmProbs
SysLocation	ifSpeed	ipForwDatagrams	IcmpInSrcQuenchs
SysServices	ifPhysAddress	ipInUnknownProtos	IcmpInRedirects
	ifAdminStatus	ipInDiscards	IcmpInEchos
	ifOperStatus	ipInDelivers	IcmpInEchoReps
	ifLastChange	ipOutRequests	IcmpInTimestamps
	ifInOctets	ipOutDiscards	IcmpTimestampReps
	ifInUcastPkts	ipOutNoRoutes	IcmpInAddrMasks
	ifInNUcastPkts	ipReasmTimeout	IcmpOutMsgs
	ifInDiscards	ipReasmReqds	IcmpOutErrors
	ifInErrors	ipReasmOKs	IcmpOutDestUnreachs
	ifInUnknownProtos	ipReasmFails	IcmpOutTimeExcds
	ifOutOctets	ipFragOKs	IcmpOutParmProbs
	ifOutUcastPkts	ipFragFails	IcmpOutSrcQuenchs
	ifOutNUcastPkts	ipFragCreates	IcmpOutRedirects
	ifOutDiscards	ipAdEntAddr	IcmpOutEchos
	ifOutErrors	ipAdEntIfIndex	IcmpOutEchoReps
	ifOutQLen	ipAdEntNetMask	IcmpOutTimestamps
	ifSpecific	ipAdEntBcastAddr	IcmpOutTimestampReps
		ipAdEntReasmMaxSize	IcmpOutAddrMasks

System MIB	Interfaces MIB	IP MIB	ICMP MIB
		IpNetToMediaIfIndex	IcmpOutAddrMaskReps
		IpNetToMediaPhysAddress	
		IpNetToMediaNetAddress	
		IpNetToMediaType	
		IpRoutingDiscards	

UDP MIB	TCP MIB	SNMP MIB
UdpInDatagrams	tcpRtoAlgorithm	snmpInPkts
UdpNoPorts	tcpRtoMin	snmpOutPkts
UdpInErrors	tcpRtoMax	snmpInBadVersions
UdpOutDatagrams	tcpMaxConn	snmpInBadCommunityNames
UdpLocalAddress	tcpActiveOpens	snmpInASNParseErrs
UdpLocalPort	tcpPassiveOpens	snmpInTooBigs
	tcpAttempFails	snmpInNoSuchNames
Address Translation MIB	tcpEstabResets	snmpInBadValues
AtIfIndex	tcpCurrEstab	snmpInReadOnlys
AtPhysAddress	tcpInSegs	snmpInGenErrs
AtNetAddress	tcpOutSegs	snmpInTotalReqVars